



UNIVERSIDAD DE CUENCA  
FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA EN ELECTRÓNICA Y  
TELECOMUNICACIONES

**DISEÑO E IMPLEMENTACIÓN DE UN ALGORITMO DE CONTROL  
ADAPTATIVO TOLERANTE A FALLOS ACTIVO BASADO EN REDES  
NEURONALES Y LÓGICA DIFUSA**

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE:  
**INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**

**Autores:**

Edisson Andrés Naula Duchi  
C.I:0105709679

Andrés Fernando García Calle  
C.I:0104647532

**Director:**

Ing. Luis Ismael Minchala Ávila, PhD.  
C.I:0301453486

CUENCA-ECUADOR  
OCTUBRE, 2017



# Resumen

Este trabajo presenta la metodología de diseño de dos controladores proporcional-integral-derivativo (PID) tolerantes a fallas. El primer controlador propone una estrategia de control tolerante a fallas pasiva (PFTC, por sus siglas en inglés) a través del cálculo en línea de los parámetros del controlador usando una red neuronal ADALINE cuya tasa de aprendizaje se ajusta con un sistema de inferencia difuso. El segundo controlador es un PID de ganancia difusa programada, cuyos rangos de variación de parámetros se ajustan en línea con información proveniente de un módulo de detección y diagnóstico de fallas (FDD, por sus siglas en inglés). La arquitectura de este esquema es la de un controlador tolerante a fallas activo (AFTC, por sus siglas en inglés). La detección y diagnóstico de fallas se realiza mediante ecuaciones de paridad. La validación de los algoritmos de control se realiza en un sistema de múltiples entradas y múltiples salidas (MIMO, por sus siglas en inglés), de tres tanques de agua interconectados en cascada, que incluye el modelo de fugas y obstrucciones. Las simulaciones permiten comparar los el desempeño de los enfoques propuestos con el algoritmo de control PID clásico.

**Palabras claves:** detección y diagnóstico de fallas, ecuaciones de paridad, espacio de paridad, PID, control tolerante a fallas, sistema difuso, red neuronal, observador.



# Abstract

This work presents the design methodology of two fault tolerant proportional-integral-derivative (PID) controllers. The first controller proposes a passive fault tolerant control strategy (PFTC) through online calculation of controller parameters using an ADALINE neural network whose learning rate is adjusted with a diffuse inference system. The second controller is a Fuzzy Gain Scheduling PID whose ranges of variation of parameters are adjusted online with the information from a fault detection and diagnostic (FDD) module. The proposed scheme's architecture forms an active fault tolerant controller (AFTC). The detection and diagnosis of faults is achieved by parity equations. The validation of the control algorithms is performed in a multiple-input multiple-output system (MIMO) of three tanks, including the leakages and obstructions models. The simulations allow us to compare the performance of the proposed approaches with the classic PID control algorithm.

**Keywords:** fault detection and diagnosis, parity equations, PID, fault tolerant control, fuzzy system, neural network, state observer



# Índice general

Resumen . . . . .	I
Abstract . . . . .	II
Índice general . . . . .	IV
Índice de figuras. . . . .	VI
Índice de tablas. . . . .	VII
Dedicatoria . . . . .	XII
Agradecimientos . . . . .	XIII
<b>1. Introducción</b>	<b>1</b>
1.1. Identificación del problema . . . . .	1
1.2. Justificación . . . . .	1
1.3. Alcance . . . . .	2
1.4. Objetivos . . . . .	2
1.4.1. Objetivo general . . . . .	2
1.4.2. Objetivos específicos . . . . .	2
<b>2. Estado del arte</b>	<b>3</b>
<b>3. Modelados de planta</b>	<b>4</b>
3.1. Motor CC . . . . .	4
3.2. Reactor de tanque agitado continuo (CSTR) . . . . .	4
3.3. Sistema de 3 tanques interconectados . . . . .	6
3.3.1. Ecuaciones dinámicas . . . . .	6
3.3.2. Modelado de fallas . . . . .	7
3.3.3. Modelado del sistema con fallas . . . . .	8
<b>4. Algoritmos de control</b>	<b>12</b>
4.1. Control proporcional integral derivativo . . . . .	12
4.1.1. Control proporcional . . . . .	12
4.1.2. Control PI . . . . .	12
4.1.3. Control proporcional derivativo . . . . .	13
4.2. Control predictivo por modelo . . . . .	13
4.2.1. Modelo predictivo . . . . .	13
4.2.2. Función objetivo . . . . .	13
4.2.3. Las reglas de control . . . . .	14
4.3. Control ADALINE PID . . . . .	14
4.3.1. Estructura del control neuronal . . . . .	14
4.3.2. Sistema de inferencia difuso . . . . .	15



4.4. Controlador de ganancia programada . . . . .	17
<b>5. Técnicas de detección y diagnóstico de fallas</b>	<b>19</b>
5.1. Ecuaciones de paridad . . . . .	19
5.1.1. Ecuaciones de paridad en tiempo discreto . . . . .	19
5.1.2. Diseño de residuos usando SVD . . . . .	22
5.2. Observadores . . . . .	23
5.3. Modelado difuso . . . . .	25
5.4. Modelado neuronal . . . . .	26
<b>6. Sistema FDD</b>	<b>28</b>
6.1. Detección . . . . .	28
6.2. Diagnóstico . . . . .	28
6.3. Estimador de fallas . . . . .	28
6.3.1. Optimizadores . . . . .	28
6.3.2. Ecuaciones dinámicas . . . . .	30
<b>7. Control tolerante a fallas</b>	<b>31</b>
7.1. Control Tolerante a Fallas Activo . . . . .	31
7.1.1. Control MPC con modelo post-falla . . . . .	31
7.1.2. Controlador de ganancia programada activo . . . . .	31
7.2. Controlador Tolerante a Fallas Pasivo . . . . .	32
<b>8. Simulaciones</b>	<b>33</b>
8.1. Pruebas controlador ADALINE PID . . . . .	33
8.1.1. Planta: Motor CC . . . . .	33
8.1.2. Planta: CSTR . . . . .	37
8.2. Pruebas del sistema de Detección y diagnóstico de Fallas . . . . .	41
8.3. Pruebas de los controladores . . . . .	43
8.3.1. Comparación controladores PID . . . . .	45
8.3.2. Comparación controlador MPC . . . . .	48



# Índice de figuras

3.1. Respuesta a un escalón de la planta nominal y su variación de $\pm 50\%$ en los parámetros $\omega_n$ y $\zeta$ . . . . .	5
3.2. Respuesta a un escalón de la planta nominal y su variación de $\pm 3\%$ en la energía de activación $\frac{E_a}{R}$ . . . . .	5
3.3. Representación de la planta utilizando la norma ISA-S5.1-84. . . . .	6
4.1. Funciones de membresía para el error . . . . .	16
4.2. Funciones de membresía para la salida . . . . .	16
4.3. Funciones de membresía para el signo . . . . .	16
4.4. Diagrama de bloques del controlador pasivo ADALINE PID. . . . .	17
4.5. Digrama de bloques FGSC. [1] . . . . .	17
4.6. Funciones de membresía. [1] . . . . .	18
5.1. Generación de residuos con un modelo en espacio de estados de un sistema MIMO en tiempo discreto.[2] . . . . .	20
5.2. Sistema y observador de estado en la salida[2] . . . . .	24
5.3. Modelamiento e Identificación Difusa. . . . .	25
5.4. Generación de Residuos. . . . .	26
5.5. Modelamiento neuronal . . . . .	27
6.1. Diagrama de bloques del diagnóstico de fallas . . . . .	29
6.2. Estimador de fallas . . . . .	29
6.3. ADALINE con una entrada recursiva . . . . .	29
7.1. MPC activo . . . . .	32
7.2. Control tolerante a fallas activo . . . . .	32
8.1. Respuesta en lazo cerrado de la planta nominal. . . . .	34
8.2. Índice de desempeño y consumo de energía de la planta nominal ante cambios de referencia. . . . .	34
8.3. Respuesta de la planta nominal ante presencia de perturbaciones. . . . .	35
8.4. Índice de desempeño y consumo de energía de la planta nominal ante presencia de perturbaciones. . . . .	35
8.5. Respuesta de la planta con un cambio de planta y referencia en los segundos 10 y 20. . . . .	36
8.6. Índice de desempeño y consumo de energía ante el cambio de planta y referencia en los segundos 10 y 20. . . . .	36
8.7. Respuesta en lazo cerrado del CSTR nominal. . . . .	37



8.8. Índice de desempeño y consumo de energía del CSTR nominal ante cambios de referencia. . . . .	37
8.9. Respuesta de la planta nominal ante presencia de perturbaciones. . . . .	38
8.10. Índice de desempeño y consumo de energía de la planta nominal ante presencia de perturbaciones. . . . .	38
8.11. Respuesta de la planta con un cambio de planta en los segundos 10 y 20. . . . .	39
8.12. Índice de desempeño y consumo de energía ante el cambio de planta en los segundos 10 y 20. . . . .	39
8.13. Saturación de actuadores en el motor de CC. . . . .	40
8.14. Residuos y detección de fallas. . . . .	41
8.15. Matriz de confusión del bloque FDD usando espacios de paridad. . . . .	42
8.16. Matriz de confusión del bloque FDD usando observadores. . . . .	42
8.17. Matriz de confusión del bloque FDD usando Modelamiento Neuronal. . . . .	43
8.18. Matriz de confusión del bloque FDD usando Modelamiento Difuso. . . . .	43
8.19. Diagrama de bloques para la primera prueba. . . . .	44
8.20. Prueba con un liqueo en el primer tanque en el segundo 142. . . . .	45
8.21. Prueba con un liqueo en el segundo tanque en el segundo 142. . . . .	46
8.22. Prueba con un liqueo en el tercer tanque en el segundo 142. . . . .	46
8.23. Prueba con obstrucción en el primer tanque en el segundo 142. . . . .	47
8.24. Prueba con obstrucción en el segundo tanque en el segundo 142. . . . .	47
8.25. Prueba con obstrucción en el tercer tanque en el segundo 142. . . . .	48
8.26. Prueba con liqueo en el primer tanque en el segundo 141. . . . .	48
8.27. Prueba con liqueo en el segundo tanque en el segundo 141. . . . .	49
8.28. Prueba con liqueo en el tercer tanque en el segundo 141. . . . .	49
8.29. Prueba con obstrucción en el primer tanque en el segundo 141. . . . .	49
8.30. Prueba con obstrucción en el segundo tanque en el segundo 141. . . . .	50
8.31. Prueba con obstrucción en el tercer tanque en el segundo 141. . . . .	50

# Índice de tablas

4.1. Reglas Difusas . . . . .	15
4.2. Reglas Difusas para $K_p'$ [1] . . . . .	18
4.3. Reglas Difusas para $T_i'$ [1] . . . . .	18
8.1. Resultados con cambios de referencia en el motor CC. . . . .	33
8.2. Presencia de perturbaciones en el motor CC. . . . .	35
8.3. Variación de parámetros y referencias en el motor CC. . . . .	36
8.4. Resumen de resultados ante cambios de referencia en el CSTR. . . . .	38
8.5. Presencia de perturbaciones en el CSTR. . . . .	38
8.6. Cambio de parámetros en el CSTR. . . . .	39
8.7. Parámetros del sistema de tres tanques interconectados. . . . .	40



Yo, Edison Andrés Naula Duchi, autor del trabajo de titulación “Diseño e implementación de un algoritmo de control adaptativo tolerante a fallos activo basado en redes neuronales y lógica difusa”, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 23 de Octubre de 2017

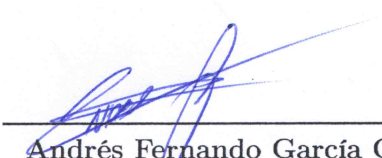
---

**Edison Andrés Naula Duchi**  
010570967-9



Yo, Andrés Fernando García Calle, autor del trabajo de titulación "Diseño e implementación de un algoritmo de control adaptativo tolerante a fallos activo basado en redes neuronales y lógica difusa", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 23 de Octubre de 2017

  
\_\_\_\_\_  
Andrés Fernando García Calle

010464753-2



Yo, Edison Andrés Naula Duchi, autor del trabajo de titulación “Diseño e implementación de un algoritmo de control adaptativo tolerante a fallos activo basado en redes neuronales y lógica difusa”, certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor/a. patrimoniales como autor.

Cuenca, 23 de Octubre de 2017

---

**Edisson Andrés Naula Duchi**  
010570967-9



Yo, Andrés Fernando García Calle, autor del trabajo de titulación “Diseño e implementación de un algoritmo de control adaptativo tolerante a fallos activo basado en redes neuronales y lógica difusa”, certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor/a.

Cuenca, 23 de Octubre de 2017

  
\_\_\_\_\_  
Andrés Fernando García Calle  
010464753-2

# Dedicatoria

Dedico esta tesis a mis padres por sus consejos, valores y por ser el pilar fundamental en todo lo que soy. A mis hermanos por haberme apoyado en todo momento y alentarme siempre a continuar. A mis maestros y amigos, quienes sin su ayuda nunca hubiera podido hacer esta trabajo.

**Edisson Andrés**

Este trabajo está dedicado a mis padres que me apoyaron en toda mi vida universitaria. A mi hermano que siempre supo como animarme cuando se presentara alguna dificultad. A mis profesores que me brindaron su conocimiento y a mis amigos que muchas veces confiaron más en mí que yo mismo.

**Andrés García**





# Agradecimientos

Quiero agradecer a mis padres, hermanos y a todos aquellos que participaron, directa o indirectamente, en la elaboración de este trabajo. A mi compañero y amigo Fernando, gracias por la perseverancia puesta. A mi director de tesis, Ing. Ismael Minchala, por su motivación y ayuda en los problemas que se presentaron en la elaboración de este trabajo. A todos los profesores que compartieron sus conocimientos, lo que fue indispensable para culminar con éxito esta carrera universitaria. Al Ing. Kenneth Palacio por su motivación por la investigación científica. Y a mis compañeros de carrera con los que he compartido buenos momentos.

**Edisson Andrés**

Agradezco primero a Dios por darme la oportunidad de estudiar en la universidad de Cuenca, a mis padres y a mi hermano que supieron motivarme en los estudios y apoyarme en todas las decisiones que he tomado. Agradezco a todos los profesores que me dieron clases, en especial al ingeniero Ismael Minchala quien me guió para realizar el presente trabajo y al ingeniero Kenneth Palacio que me contagió su entusiasmo por la investigación científica y el aprendizaje de máquina. Finalmente agradezco a mis compañeros y amigos que han sido un gran apoyo para mí, en especial a Edison que puso gran empeño en el proyecto para sobrepasar todas las dificultades que se nos presentaron.

**Andrés Fernando**



# Capítulo 1

## Introducción

### 1.1. Identificación del problema

Los sistemas tecnológicos, como eléctricos, industriales, control de aeronaves, etc., están propensos a problemas críticos de seguridad y operatividad; por ello dependen de sistemas de control complejos. Los controladores clásicos están bien definidos para garantizar la estabilidad y el rendimiento deseado alrededor de un punto operativo. Sin embargo, estos esquemas de control son incapaces de mantener un rendimiento deseado en escenarios con fallas. Una falla es definida como el cambio de un parámetro fuera de los márgenes de tolerancias, es decir, una situación externa al sistema que modifica su respuesta. Las fallas son del tipo multiplicativas si un parámetro interno del sistema es modificado. Las fallas son del tipo aditivas si pueden representarse como entradas adicionales al sistema. Una falla puede estar ubicada en sensores, actuadores o componentes. El comportamiento de las fallas en el tiempo las divide en tres tipos: abruptas, incipientes e intermitentes. Las fallas consideradas en este trabajo son abruptas.

- Fallas abruptas cuando se manifiesta como un cambio repentino, se modelan como una función tipo escalón.
- Fallas incipientes si aumentan su magnitud con respecto al tiempo, se modelan como una función rampa.
- Fallas intermitentes si aparecen y desaparecen sin un patrón determinado.

### 1.2. Justificación

Es necesario diseñar sistemas de control tolerante a fallas (FTC, por sus siglas en inglés), que proporcionen un rendimiento razonable. Los sistemas PFTC representan una mejora a los controladores clásicos, pero son diseñados para soportar el efecto de fallas específicas. Una alternativa para diseñar un PFTC es mediante controladores adaptativos [3]. Los controladores clásicos suelen trabajar hasta cierto nivel de falla o suelen tener respuestas menos eficientes ante estas.



## 1.3. Alcance

Este trabajo se presentan dos tipos de control tolerante a fallos: el primero es un PFTC basado en redes neuronales y un sistema difuso para la adaptación de la tasa de aprendizaje; el segundo controlador es un AFTC, que combina un controlador PID de ganancia programada y un sistema FDD basado en ecuaciones de paridad. Se implementa una simulación de un sistema MIMO para las pruebas y para implementar el algoritmo FTC.

## 1.4. Objetivos

### 1.4.1. Objetivo general

Desarrollar e implementar un algoritmo AFTC en tiempo real y un sistema FDD utilizando las técnicas de inteligencia artificial, redes neuronales y lógica difusa.

### 1.4.2. Objetivos específicos

- Diseñar un algoritmo de control adaptativo neuronal en combinación con lógica difusa para la actualización en tiempo real de los parámetros del controlador.
- Diseñar un sistema FDD utilizando ecuaciones de paridad.
- Simular el algoritmo propuesto en un sistema de tanques interconectados.

## Capítulo 2

### Estado del arte

Un control adaptativo puede ser diseñado a partir de uno tradicional. En [4] se tiene un control proporcional integral (PI) como base para agregarle adaptabilidad. El control PI modifica sus parámetros por medio de un programador de ganancias en base a la señal de error y se complementa con un sistema FDD que modifica las entradas y salidas del controlador. El bloque FDD utiliza residuos entre la planta y modelos difusos que incluyen la dinámica de fallas.

La información del FDD se puede utilizar de varias maneras para obtener un ATFC. En [5] existe un gobernador que modifica las referencias del sistema de acuerdo a las fallas que se presenten. Las señales de referencia son degradadas para que los actuadores no se saturen ni se dañen. Para no desestabilizar al sistema el cambio de referencia se lo realiza de forma gradual.

Los residuos del FDD también pueden obtenerse mediante varios enfoques. El diseño de modelos redundantes o modelos que incluyan la dinámica de una falla es uno de estos enfoques. En [5] se tiene un modelo linealizado de la planta sin falla y otro modelo con una degradación que simula las fallas, los residuos se calculan con ambos modelos y con esto se diagnostica si existe o no falla.

El modelamiento con Takagi-Sugeno (T-S) es una buena opción para modelar procesos no lineales ([6], [5], [7]). En [4] los modelos difusos se obtuvieron con T-S, cada modelo incluye la dinámica de una falla diferente. El residuo generado por cada modelo va hacia un evaluador y se determina si es una falla o no. Observadores de estados también son utilizados para la obtención de residuos, como es el caso de [7]. Las fallas buscadas se presentan en los actuadores y sensores del sistema.

En [8] se propone control adaptativo que tiene como base un controlador PID. El control adaptativo simula un control PID cuyos parámetros son los pesos de 3 entradas de un ADALINE. La información de entrada del ADALINE son las 3 últimas muestras del error. El algoritmo de actualización de pesos del ADALINE (aprendizaje) actúa cuando existe un cambio en la señal de error.

Los trabajos en [9] y [10] proponen un control adaptativo ADALINE que simula los parámetros proporcional, integral y derivativo de un PID; modificándolos en base al error absoluto. [11] usa una sola neurona como controlador. La salida de la neurona tiene una ganancia controlada por un sistema difuso, también mejora el entrenamiento de la red con un ajuste de la tasa de aprendizaje.

# Capítulo 3

## Modelados de planta

### 3.1. Motor CC

El primer modelo utilizado es una planta experimental de un motor de CC. Este tiene una ecuación de segundo orden en el dominio de Laplace (Ec. 3.1), con un retardo  $\theta$ .

$$G(s) = \frac{\omega_n^2 e^{-\theta s}}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (3.1)$$

Los cambios en la planta se modelan como una variación del 50 % de los parámetros, como frecuencia natural  $\omega_n$  y la constante de amortiguamiento  $\zeta$ , para así modificar la respuesta de la planta. Las ecuaciones del modelo y sus variaciones se muestran en (Ec.3.2).

$$\begin{aligned} G(s) &= \frac{122,5e^{-0,1s}}{s^2 + 30,66s + 94,16} \\ G_{-50\%}(s) &= \frac{313,6e^{-0,1s}}{s^2 + 19,62s + 241} \\ G_{+50\%}(s) &= \frac{30,63e^{-0,1s}}{s^2 + 23s + 23,64} \end{aligned} \quad (3.2)$$

La respuesta a un escalón unitario, de la planta y sus variaciones, se pueden observar en la Fig.3.1. Se observa que las magnitud

### 3.2. Reactor de tanque agitado continuo (CSTR)

Un CSTR es una planta con un comportamiento no lineal y amplios rangos de funcionamiento. El modelo utilizado, obtenido de [12], está representado por las ecuaciones (Ec.3.3). En estas se modela el comportamiento de la concentración de material  $a$  ( $C_a$ ) y la temperatura del reactor ( $T$ ). La variable de control es la temperatura de la chaqueta

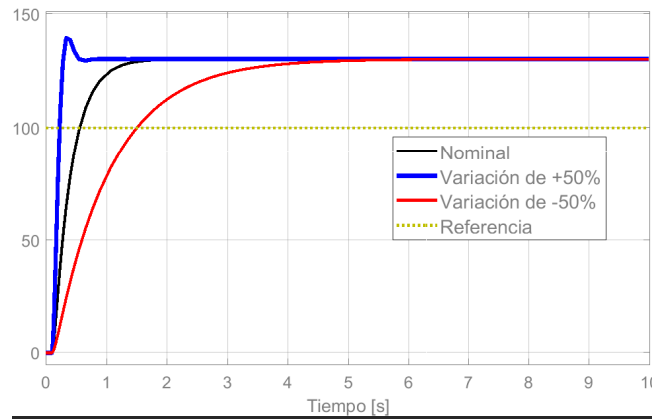


Figura 3.1: Respuesta a un escalón de la planta nominal y su variación de  $\pm 50\%$  en los parámetros  $\omega_n$  y  $\zeta$ .

$(T_j)$ .

$$\begin{aligned} \frac{dC_a}{dt} &= \left(\frac{q}{V}\right) \cdot (C_{af} - C_a) - k_o \cdot \exp\left[-\frac{E_a}{RT}\right] \\ \frac{dT}{dt} &= \left(\frac{q}{V}\right) \cdot (T_f - T) - \frac{\Delta H}{\rho C_p} \cdot \left[ k_o \cdot \exp\left[-\frac{E_a}{RT}\right] \cdot C_a \right. \\ &\quad \left. - \left(\frac{U \cdot A}{\rho C_p V}\right) \cdot (T - T_j) \right] \end{aligned} \quad (3.3)$$

Los cambios en la planta se modelan como una variación en el término de la energía de activación  $\frac{E_a}{R}$  de  $\pm 3\%$ . La respuesta a un escalón unitario, de la planta original y sus variaciones, se observa en la Fig. 3.2.

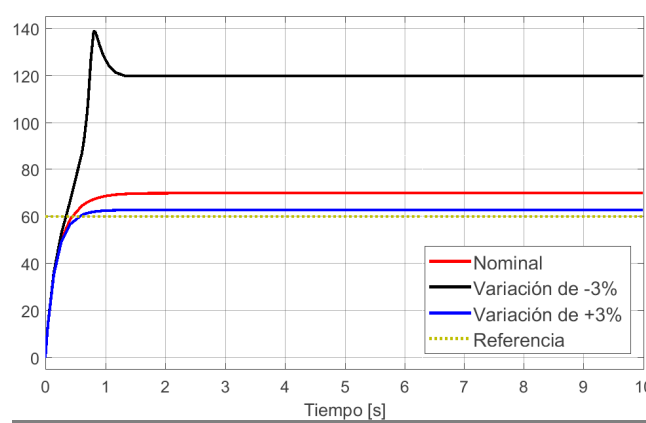


Figura 3.2: Respuesta a un escalón de la planta nominal y su variación de  $\pm 3\%$  en la energía de activación  $\frac{E_a}{R}$ .



### 3.3. Sistema de 3 tanques interconectados

El control de nivel de liquido en un tanque y el flujo entre tanques es un problema común en procesos industriales. Normalmente se requiere bombear liquido en un tanque, almacenarlo y luego bombear liquido a otro tanque. El cambio de nivel en el tanque se modela con no linealidades cuadráticas. Este sistema permite probar el rendimiento de los controladores propuestos en un sistema MIMO acoplado y no lineal.

#### 3.3.1. Ecuaciones dinámicas

La Fig.3.3 presenta el diagrama de instrumentación del sistema multi-tanques. Las salidas del sistema corresponden al nivel de agua en cada tanque. Las variables controlables son:

- Flujo de agua que alimenta al primer tanque.
- Flujo de agua que alimenta al tercer tanque.
- Apertura de la válvula que conecta los tanques 1 y 2.

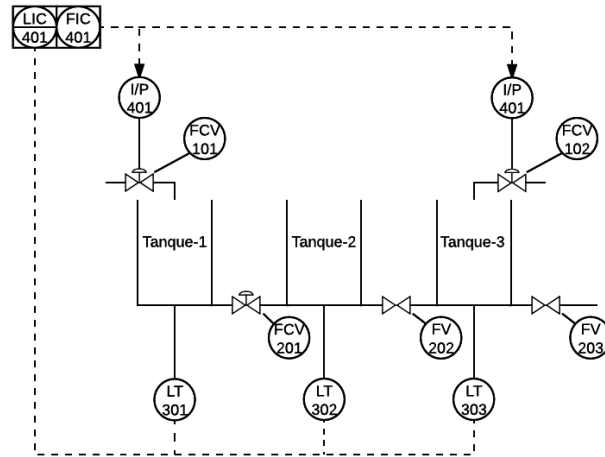


Figura 3.3: Representación de la planta utilizando la norma ISA-S5.1-84.

Las ecuaciones diferenciales que representan al sistema son:

$$\begin{aligned}
 \frac{dh_1}{dt} &= \frac{1}{S_1} \cdot \left( f_1 - A \cdot \sigma \cdot \sqrt{2g \cdot (h_1 - h_2)} \right) \\
 \frac{dh_2}{dt} &= \frac{1}{S_2} \cdot \left( A \cdot \sigma \cdot \sqrt{2g \cdot (h_1 - h_2)} - A \cdot \sqrt{2g \cdot (h_2 - h_3)} \right) \\
 \frac{dh_3}{dt} &= \frac{1}{S_3} \cdot \left( 3 + A \cdot \sqrt{2g \cdot (h_2 - h_3)} - A \sqrt{2g \cdot h_3} \right)
 \end{aligned} \tag{3.4}$$

Siendo:

- $f_1$  : flujo de agua entrante en el tanque 1.

- $f_3$  : flujo de agua entrante en el tanque 3.
- $\sigma$  : coeficiente de apertura de la válvula controlable.
- $A$  : área transversal del tubo que interconecta cada tanque.
- $S_1, S_2, S_3$  : sección de área de cada tanque.
- $h_1, h_2, h_3$  : nivel de agua de cada tanque.

### 3.3.2. Modelado de fallas

Una falla esta representada por el cambio de un parámetro fuera de un rango de tolerancia. Existen dos tipos de fallas: las fallas multiplicativas surgen al considerar un cambio en un parámetro del proceso, también es la forma más común de modelar. Las fallas aditivas son modeladas como entradas adicionales al proceso. Una falla puede estar ubicada en sensores, actuadores o componentes.

El comportamiento de las fallas, en el tiempo, las divide en tres tipos: Una falla es abrupta cuando se manifiesta como un cambio repentino y se modela como una función tipo escalón. Una falla incipiente aumenta su nivel con respecto al tiempo y se modela como una función rampa. Una falla es intermitente cuando aparece y desaparece sin un patrón determinado. Las fallas consideradas en el trabajo son abruptas.

#### Fugas de agua en los tanques

Esta falla se modela como la pérdida de agua debido a un agujero en el fondo de un tanque. Se representa de la siguiente forma:

$$\begin{aligned}\frac{dh_1}{dt} &= \frac{1}{S_1} \cdot \left( f_1 - A \cdot \sigma \cdot \sqrt{2g \cdot (h_1 - h_2)} - \alpha_1 \cdot \sigma \cdot \sqrt{2g \cdot (h_1 - h_2)} \right) \\ \frac{dh_2}{dt} &= \frac{1}{S_2} \cdot \left( A \cdot \sigma \cdot \sqrt{2g \cdot (h_1 - h_2)} - A \cdot \sqrt{2g \cdot (h_2 - h_3)} - \alpha_2 \cdot \sqrt{2g \cdot (h_2 - h_3)} \right) \\ \frac{dh_3}{dt} &= \frac{1}{S_3} \cdot \left( f_3 + A \cdot \sqrt{2g \cdot (h_2 - h_3)} - A \sqrt{2g \cdot h_3} - \alpha_3 \cdot \sqrt{2g \cdot (h_2 - h_3)} \right)\end{aligned}\quad (3.5)$$

Donde los términos  $\alpha_{1,2,3}$  corresponden al radio del agujero que provoca pérdida de agua en el tanque correspondiente.

#### Obstrucción en las interconexiones y en la salida del tercer tanque

Una obstrucción supone un cambio en la sección de las interconexiones. Esto se representa de la siguiente forma:

$$\begin{aligned}\frac{dh_1}{dt} &= \frac{1}{S_1} \cdot \left( f_1 - A(1 - \beta_1) \cdot \sigma \cdot \sqrt{2g \cdot (h_1 - h_2)} \right) \\ \frac{dh_2}{dt} &= \frac{1}{S_2} \cdot \left( A(1 - \beta_1) \cdot \sigma \cdot \sqrt{2g \cdot (h_1 - h_2)} - A(1 - \beta_2) \cdot \sqrt{2g \cdot (h_2 - h_3)} \right) \\ \frac{dh_3}{dt} &= \frac{1}{S_3} \cdot \left( f_3 + A(1 - \beta_2) \cdot \sqrt{2g \cdot (h_2 - h_3)} - A(1 - \beta_3) \sqrt{2g \cdot h_3} \right)\end{aligned}\quad (3.6)$$

Donde los valores de  $\beta_{1,2,3}$  corresponden a las variaciones en la sección de área de las conexiones y la salida.

### Fallas en las válvulas de entrada de flujo

Esta falla de actuador se representa como una variación en el flujo de agua que proporcionan las válvulas de entrada. Para representar esta falla se utilizan las siguientes ecuaciones:

$$\begin{aligned}\frac{dh_1}{dt} &= \frac{1}{S_1} \cdot \left( f_1(1 - \gamma_1) - A \cdot \sigma \cdot \sqrt{2g \cdot (h_1 - h_2)} \right) \\ \frac{dh_2}{dt} &= \frac{1}{S_2} \cdot \left( A \cdot \sigma \cdot \sqrt{2g \cdot (h_1 - h_2)} - A \cdot \sqrt{2g \cdot (h_2 - h_3)} \right) \\ \frac{dh_3}{dt} &= \frac{1}{S_3} \cdot \left( f_3(1 - \gamma_3) + A \cdot \sqrt{2g \cdot (h_2 - h_3)} - A \sqrt{2g \cdot h_3} \right)\end{aligned}\quad (3.7)$$

Donde los valores de  $\gamma_{1,2,3}$  corresponden a las variaciones de cantidad de flujo de agua que proporciona las válvulas de entrada.

### 3.3.3. Modelado del sistema con fallas

Las ecuaciones que representan un sistema en espacio de estados (EE) se muestran a continuación:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du}\end{aligned}\quad (3.8)$$

Siendo para este artículo  $\mathbf{y} = [h_1, h_2, h_3]^T$  y  $\mathbf{u} = [f_1, f_3, \sigma]^T$  las salidas y entradas del sistema. Las matrices de estados se calculan de la siguiente manera:

$$\begin{aligned}\mathbf{A} &= \begin{bmatrix} -\frac{T_1}{S_1} & \frac{T_1}{S_1} & 0 \\ \frac{T_1}{S_2} & -\frac{T_2}{S_2} & \frac{T_2}{S_2} \\ 0 & \frac{T_2}{S_3} & -\frac{T_3}{S_3} \end{bmatrix} \\ \mathbf{B} &= \begin{bmatrix} \frac{1}{S_1} & 0 & -\frac{T_4}{S_1} \\ 0 & 0 & \frac{T_4}{S_2} \\ 0 & \frac{1}{S_3} & 0 \end{bmatrix} \\ \mathbf{C} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \mathbf{D} &= 0\end{aligned}$$

Considerando  $g = 9,81 \frac{m}{s^2}$ , se tiene que:

- $T1 = \frac{Ag\sigma}{\sqrt{2g(h_1 - h_2)}}$
- $T2 = \frac{Ag}{\sqrt{2g(h_2 - h_3)}}$

- $T3 = \frac{Ag}{\sqrt{2g \cdot h_3}}$
- $T4 = A\sqrt{2g(h_1 - h_2)}$

El paso previo a generar residuos es modelar las fallas. El modelado de las fallas se puede observar en la sección 3.3.2. Las fallas están representadas en el espacio de estados, similar a (3.8), como un efecto aditivo a la planta sin fallas.

$$\begin{aligned}\dot{x} &= \mathbf{A}x + \mathbf{B}u + \mathbf{F}f \\ y &= \mathbf{C}x + \mathbf{D}u\end{aligned}\tag{3.9}$$

$$\begin{aligned}\dot{x} &= \mathbf{A}x + \mathbf{B}u + F_l f_l + F_o f_o + F_a f_a \\ y &= \mathbf{C}x + \mathbf{U}u\end{aligned}\tag{3.10}$$

En este caso el vector  $F$  representa el efecto que tiene la falla  $f(t)$  en la planta. Se busca obtener un modelo como en (3.10), donde los efectos de las fallas son aditivos al modelo sin fallas.

### Fugas en los tanques.

Las fugas presentan una disminución de líquido en el nivel de los tanques. Su representación, en espacio de estados, se deduce al linealizar sus ecuaciones diferenciales. El conjunto de ecuaciones (3.11) representan las fugas del modelo planteado en espacio de estados.

$$\begin{aligned}\mathbf{A} &= \begin{bmatrix} -\frac{T1(1-\alpha_1/Ag)}{S_1} & \frac{T1}{S_1} & 0 \\ \frac{T1}{S_2} & -\frac{T2(1-\alpha_2/Ag)}{S_2} & \frac{T2}{S_2} \\ 0 & \frac{T2}{S_3} & -\frac{T3(1-\alpha_3/Ag)}{S_3} \end{bmatrix} \\ \mathbf{B} &= \begin{bmatrix} \frac{1}{S_1} & 0 & -\frac{T4}{S_1} \\ 0 & 0 & \frac{T4}{S_2} \\ 0 & \frac{1}{S_3} & 0 \end{bmatrix} \\ \mathbf{C} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \mathbf{D} &= 0\end{aligned}\tag{3.11}$$

El efecto de las fugas altera únicamente los valores de la matriz  $A$ . Con estas matrices se requiere expresar, el efecto de las fugas, como un efecto aditivo al modelo sin fallas y obtener un espacio de paridad de las fallas.

$$\begin{aligned}F_l &= \begin{bmatrix} -\frac{T1/Ag}{S_1} & \frac{T1}{S_1} & 0 \\ \frac{T1}{S_2} & -\frac{T2/Ag}{S_2} & \frac{T2}{S_2} \\ 0 & \frac{T2}{S_3} & -\frac{T3/Ag}{S_3} \end{bmatrix} \\ f_l &= \vec{\alpha} \cdot x\end{aligned}\tag{3.12}$$

### Obstrucción en las interconexiones y en la salida del tercer tanque.

Las obstrucciones, al igual que las fugas, son modeladas como un cambio en la sección de las interconexiones de los tanques. Su representación en espacio de estados es (3.13).

$$\begin{aligned}
 A &= \begin{bmatrix} -\frac{T1(1-\beta_1)}{S_1} & \frac{T1}{S_1} & 0 \\ \frac{T1}{S_2} & -\frac{T2(1-\beta_2)}{S_2} & \frac{T2}{S_2} \\ 0 & \frac{T2}{S_3} & -\frac{T3(1-\beta_3)}{S_3} \end{bmatrix} \\
 B &= \begin{bmatrix} \frac{1}{S_1} & 0 & -\frac{T4(1-\beta_1)}{S_1} \\ 0 & 0 & \frac{T4(1-\beta_1)}{S_2} \\ 0 & \frac{1}{S_3} & 0 \end{bmatrix} \\
 C &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 D &= 0
 \end{aligned} \tag{3.13}$$

El efecto de las obstrucciones está presente en las matrices  $AyB$ , pero al igual que en las fugas se puede obtener una expresión como en (3.9) que se deduce a continuación:

$$\begin{aligned}
 F_o \cdot f_o &= \begin{bmatrix} \frac{T1\beta_1(x_1-x_2)}{S_1} + \frac{T4\beta_1u_3}{S_1} \\ -\frac{T1\beta_1}{S_2} + \frac{T2\beta_2(x_2-x_3)}{S_2} - \frac{T4\beta_1u_3}{S_2} \\ \frac{T2\beta_2x_2}{S_3} + \frac{T3\beta_3x_3}{S_3} \end{bmatrix} \\
 F_o \cdot f_o &= \begin{bmatrix} \frac{1}{S_1} \\ 0 \\ 0 \end{bmatrix} \cdot f_{o1} + \begin{bmatrix} 0 \\ -\frac{1}{S_2} \\ 0 \end{bmatrix} \cdot f_{o2} + \begin{bmatrix} 0 \\ \frac{1}{S_2} \\ 0 \end{bmatrix} \cdot f_{o3} \\
 &+ \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{S_3} \end{bmatrix} \cdot f_{o4} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{S_3} \end{bmatrix} \cdot f_{o5}
 \end{aligned} \tag{3.14}$$

$$\begin{aligned}
 f_{o1} &= T1\beta_1(x_1 - x_2) + T4\beta_1u_3 \\
 f_{o2} &= T1\beta_1x_1 - T4\beta_1u_3 \\
 f_{o3} &= T2\beta_2(x_2 - x_3) \\
 f_{o4} &= T2\beta_2x_2 \\
 f_{o5} &= T3\beta_3x_3
 \end{aligned} \tag{3.15}$$

### Falla en el actuador de flujo de entrada

Si siguiendo el mismo procedimiento, aplicado en las fallas anteriores, el espacio de paridad de las fallas de los actuadores está dado por (3.16).

$$\begin{aligned} \mathbf{F}_a &= \begin{bmatrix} \frac{1}{s_1} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{s_3} \end{bmatrix} \\ f_a &= \vec{\gamma} \cdot u \end{aligned} \tag{3.16}$$

# Capítulo 4

## Algoritmos de control

### 4.1. Control proporcional integral derivativo

El control PID es muy usado en la industria debido a su fácil implementación, su alta robustez y su simplicidad [13]. Este controlador basa su funcionamiento en 3 constantes: la proporcional, la integral y la derivativa. La ecuación que representan al control PID es:

$$u(t) = K \left( e(t) + \frac{1}{T_i} \int_0^t e(\tau) \cdot d\tau + T_d \frac{de(t)}{dt} \right) \quad (4.1)$$

En donde  $e(t)$  es la diferencia entre la referencia del sistema y la salida medida del mismo.  $K$ ,  $T_i$  y  $T_d$  son las constantes proporcional, integral y derivativa respectivamente [14].

#### 4.1.1. Control proporcional

Un control proporcional tiene la siguiente ecuación:

$$u(t) = Ke(t) \quad (4.2)$$

Este control es el producto del error con la constante proporcional. El control intenta reducir el error de estado estacionario a cero. Es necesario un control proporcional integral (PI) para mejorar la reducción de error.

#### 4.1.2. Control PI

Un control PI tiene una respuesta proporcional al error y al error acumulado. Con los controles tipo PI se obtienen salidas de sistema sin error de estado estacionario. La ecuación de un control tipo PI, que es parte de la ecuación, es (Ec.4.1):

$$u = K \left( e + \frac{1}{T_i} \int_0^t e(\tau) \cdot d\tau \right) \quad (4.3)$$

### 4.1.3. Control proporcional derivativo

El control proporcional derivativo (PD) reduce el tiempo de corrección de error. La acción derivativa del control PD brinda una respuesta del sistema proporcional a la derivada del error, es decir, se obtiene una respuesta futura del sistema [14]. La ecuación de un control proporcional derivativo a partir de la ecuación (Ec.4.1) es:

$$u = K \left( e + T_d \frac{de}{dt} \right) \quad (4.4)$$

## 4.2. Control predictivo por modelo

Los controles predictivos por modelo (MPC) no usan una misma estrategia de control, de hecho existen varias estrategias. Como su nombre lo indica, el MPC necesita de un modelo para calcular respuestas futuras de este en un horizonte de predicción dado, minimizando una función objetivo [15]. Los elementos que comúnmente se utilizan en un MPC son: un modelo predictivo, una función objetivo y las reglas de control.

### 4.2.1. Modelo predictivo

La selección de un buen modelo implica un buen desempeño del MPC. El modelo debe ser capaz de replicar la dinámica del sistema para así obtener buenas predicciones, teniendo en cuenta las respuestas actuales ( $y(t+k|t)$ ). Los modelos son relaciones entre las salidas y las entradas medidas. Existen varias formas de modelar un sistema entre las cuales se pueden mencionar los siguientes:

- Modelo de convolución, secuencia ponderada o respuesta al impulso.
- Modelo de respuesta al escalón.
- Función de transferencia
- Espacio de estados
- Modelo difuso
- Modelo neuronal

### 4.2.2. Función objetivo

Para obtener las reglas o leyes de control es necesario trabajar primero con la función de costo del MPC. El objetivo del control es seguir una señal de referencia, penalizando el esfuerzo del control para lograr hacerlo. [15] nos brinda una ecuación general para la función de objetivo es la siguiente:

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} \delta(j) [\hat{y}(t+j|t) - \omega(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j) [\Delta(u)(t+j-1)]^2 \quad (4.5)$$

El segundo término de (Ec.4.5) corresponde al esfuerzo del control, que es obviado en algunas ocasiones. En la función de costo se deben tomar en cuenta:



1. Parámetros:  $N_1$  y  $N_2$  son los horizontes mínimo y máximo de predicción,  $N_u$  es el horizonte de control. Los coeficientes  $\delta(j)$  y  $\lambda(j)$  son secuencias que consideran el comportamiento futuro.
2. Trayectoria de referencia: algunas aplicaciones conocen a priori la trayectoria de referencia  $\omega(t+j)$ , dando la ventaja al controlador de poder realizar una transición suave al momento de cambios de referencia, pero esta no necesariamente debe ser la trayectoria real ya que se puede tener una aproximación con los valores conocidos.
3. Restricciones: Los procesos tienen respuestas limitadas por algún rango de valores en la práctica, como por ejemplo en una válvula se tiene una posición máxima y mínima para mantenerla abierta o cerrada, además debe poder soportar una presión máxima de funcionamiento. Estas restricciones deben ser tomadas en cuenta en la función de costo a ser minimizada.

### 4.2.3. Las reglas de control

El MPC pretende encontrar los valores para los cuales la respuesta del sistema controlado siga una referencia, estos valores  $u(t+k|t)$  deben minimizar  $J$  en la (Ec.4.5), utilizando para ello el modelo elegido y sustituyendo las respuestas  $\hat{y}(t+k|t)$  en la función de costo. Si no existen restricciones la minimización de  $J$  tiene una solución analítica, caso contrario debe utilizarse algún algoritmo de optimización.

## 4.3. Control ADALINE PID

Este controlador consta de dos partes: un control neuronal tipo ADALINE y un sistema difuso. El ADALINE calcula la señal de control que minimiza el error absoluto. Sus entradas son el error y sus dos muestras anteriores. Por otra parte, el sistema difuso ajusta la tasa de rapidez de aprendizaje del ADALINE mediante un conjunto de reglas difusas.

### 4.3.1. Estructura del control neuronal

La salida de la red ADALINE se calcula mediante la (Ec.4.6). El aprendizaje se basa en la minimización del error cuadrático medio al modificar cada peso  $\omega_i$  de manera proporcional a la derivada del error respecto al peso.

$$y = \sum_{i=1}^n x_i \omega_i + \phi \quad (4.6)$$

Sustituyendo las entradas  $x_i$  de la (Ec.4.6) por cada muestra de error, y fijando a  $\phi$  como la señal de control anterior, se obtiene:

$$u_k = u_{k-1} + \omega_1 e_k + \omega_2 e_{k-1} + \omega_3 e_{k-2} \quad (4.7)$$

La (Ec.4.7) se asemeja a la ecuación de un PID clásico, en donde  $\omega_1, \omega_2$  y  $\omega_3$  representan a los parámetros integral, proporcional y derivativo respectivamente. Las señales de control y la actualización de pesos se calculan usando la (Ec.4.8) donde,  $E =$

$[e_k \ e_{k-1} \ e_{k-2}]$  es el vector de error,  $W = [\omega_1 \ \omega_2 \ \omega_3]$  el vector de pesos y  $\eta$  la tasa de aprendizaje de la red.

$$\begin{aligned} W_{new} &= \eta e_k E + W_{old} \\ u_k &= W_{new}^T E + u_{k-1} \end{aligned} \quad (4.8)$$

### 4.3.2. Sistema de inferencia difuso

En la (Ec.4.8) la tasa de aprendizaje  $\eta$  puede ser un valor arbitrario, el cual pondera el aporte del error en el aprendizaje. Esta tasa de aprendizaje manipula la velocidad de convergencia del algoritmo (4.8). Por tanto,  $\eta$  ofrece un grado de libertad extra en el diseño de esta estrategia de control. Un sistema difuso se encarga del cálculo de  $\eta$  a través del procesamiento de señales de error. Las entradas del sistema difuso son: el error relativo, que se calcula mediante la (Ec.4.9), y el signo del error.

$$e_k = \frac{r(k) - y(k)}{r(k)} \quad (4.9)$$

El método de defusificación empleado, para la salida del sistema difuso, es el del centroide (Ec.4.10), donde  $R$  es el numero reglas,  $x$  las entradas y  $\mu_x(x)$  las funciones de membresía activas para cada entrada  $x$ . La tasa de aprendizaje se calcula como:

$$f_a = \frac{\sum_{x=1}^R x \mu_x(x)}{\sum_{x=1}^R \mu_x(x)} \quad (4.10)$$

$$\eta = 0,00025 f_a \quad (4.11)$$

La Tabla I muestra el conjunto de reglas del sistema difuso. Se considera la propuesta de [16].

Tabla 4.1: Reglas Difusas

		Sign	
		P	N
Error	G	MP	MN
	M	PN	PP
	P	N	P
	Z	Z	Z

Las funciones de membresía usadas para el error y la salida se muestran en las Fig.4.1 y Fig.4.2 respectivamente.

La función de membresía del signo se representa en la Fig.4.3. Cabe destacar que para esta función de membresía los valores que pueden existir son -1, 0 y 1. Se utiliza esta función para simplificar las reglas del sistema difuso.

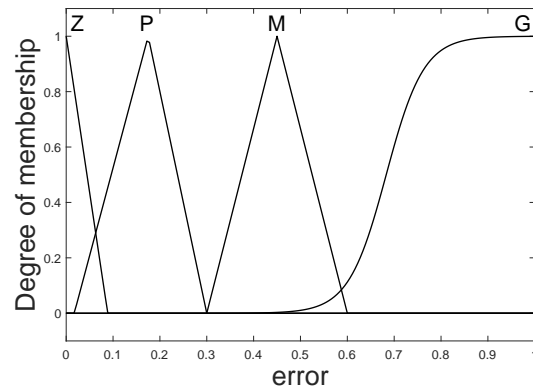


Figura 4.1: Funciones de membresía para el error

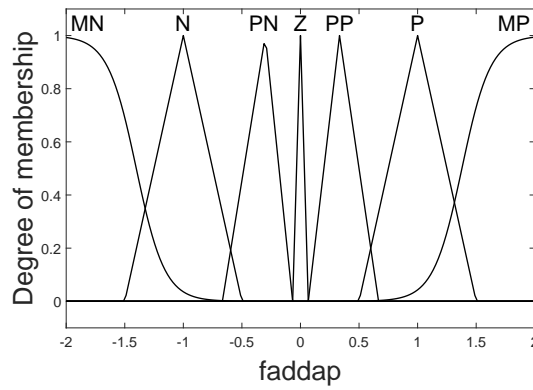


Figura 4.2: Funciones de membresía para la salida

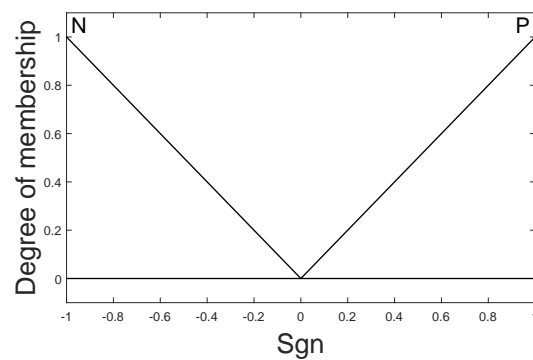


Figura 4.3: Funciones de membresía para el signo

La Fig. 4.4 muestra el diagrama de bloques del sistema de control PFTC propuesto.

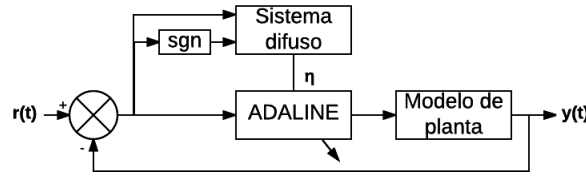


Figura 4.4: Diagrama de bloques del controlador pasivo ADALINE PID.

## 4.4. Controlador de ganancia programada

El esquema de programación propuesto por [1] explota el sistema de inferencia difusa para adaptar en línea los valores de las ganancias proporcional e integral de un controlador PI como se observa en la Fig.4.5.

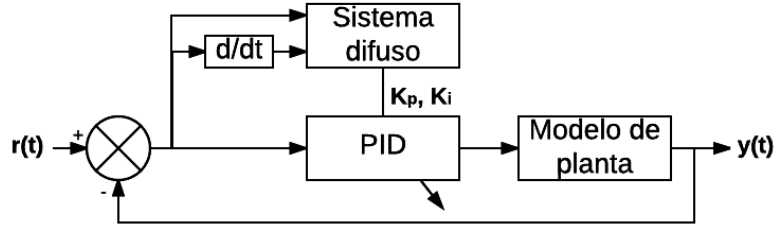


Figura 4.5: Diagrama de bloques FGSC. [1]

Los valores proporcionados por el sistema difuso son usados como factores de corrección de los valores de  $K_p$  y  $T_i$  usando la (Ec.4.12) en donde los valores de  $K'_p$  y  $T'_i$  están normalizados.

$$\begin{aligned} K_p &= (K_p^{max} - K_p^{min}) K'_p + K_p^{min} \\ T_i &= (T_i^{max} - T_i^{min}) T'_i + T_i^{min} \end{aligned} \quad (4.12)$$

Los valores de  $K'_p$  y  $T'_i$  son determinados mediante un conjunto de reglas *if-then* con una forma

$$\text{Rule}_n : \text{if } e(k) \text{ and } \dot{e}(k) \text{ is } B_n, \text{ then } K'_p \text{ is } C_n \text{ and } T'_i \text{ is } D_n \quad (4.13)$$

donde  $A_n$ ,  $B_n$  y  $C_n$  (con  $n = 1, 2, \dots, M$ ) son conjuntos difusos que corresponden a  $K'_p$  y  $T'_i$ . La forma de las funciones de membresía se observan en la Fig.4.6.

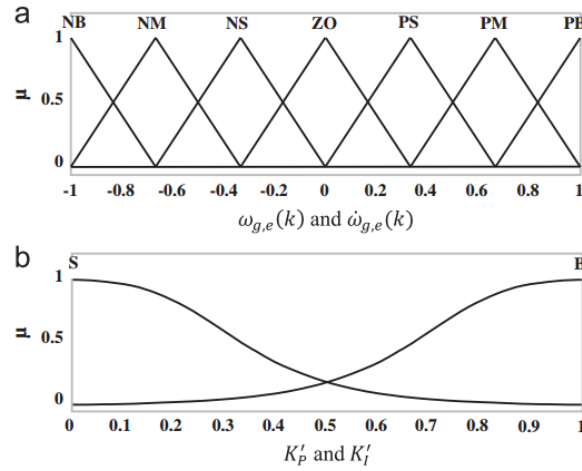


Figura 4.6: Funciones de membresía. [1]

Tabla 4.2: Reglas Difusas para  $K'_p$  [1]

		ep(k)						
		NB	NM	NS	ZO	PS	PM	PB
e(k)	NB	B	B	B	B	B	B	B
	NM	S	B	B	B	B	B	S
	NS	S	S	B	B	B	S	S
	ZO	S	S	S	B	S	S	S
	PS	S	S	B	B	B	S	S
	PM	S	B	B	B	B	B	S
	PB	B	B	B	B	B	B	B

Tabla 4.3: Reglas Difusas para  $T'_i$  [1]

		ep(k)						
		NB	NM	NS	ZO	PS	PM	PB
e(k)	NB	B	B	B	B	B	B	B
	NM	B	B	S	S	S	B	B
	NS	B	B	B	S	B	B	B
	ZO	B	B	B	S	B	B	B
	PS	B	B	B	S	B	B	B
	PM	B	B	S	S	S	B	B
	PB	B	B	B	B	B	B	B

Las reglas base utilizadas se observan en las tablas 4.2 y 4.3 y fueron formuladas en base a *expert's knowledge*. En [1] se propone el operador lógico *and* con el operador mínimo y el método de defusificación usado es del centroide.

# Capítulo 5

## Técnicas de detección y diagnóstico de fallas

### 5.1. Ecuaciones de paridad

Una manera directa de detectar fallas en proceso es comparar el comportamiento del proceso con un modelo del proceso. Este modelo describe el comportamiento nominal, es decir sin fallas en el proceso. Las diferencias de señales entre el proceso y el modelo se ven reflejadas en los residuos, por ende en estos podemos buscar coherencia con las fallas. El diseño de residuos puede hacerse con funciones de transferencia y en espacio de estados. En esta sección se diseñan residuos usando espacios de estados en tiempo discreto.

#### 5.1.1. Ecuaciones de paridad en tiempo discreto

Las ecuaciones de paridad en espacio de estados en tiempo discreto son fáciles de derivar e implementar que en tiempo continuo. Según la Fig.5.1 el proceso está representado por las ecuaciones (5.1)

$$\begin{aligned}x(k+1) &= \mathbf{A}x(k) + \mathbf{B}u(k) + \mathbf{V}v(k) + \mathbf{L}f(k) \\y(k) &= \mathbf{C}x(k) + \mathbf{N}n(k) + \mathbf{M}f(k)\end{aligned}\tag{5.1}$$

En donde  $v(k)$  y  $n(k)$  son señales de perturbación no medibles, pero en este trabajo no se tomarán en cuenta. Las fallas  $f(k)$  son aditivas y pueden estar compuestas por fallas aditivas en la entrada  $f_l(k)$  o en la salida  $f_m(k)$  del proceso. El diseño de los residuos se sigue el proceso planteado en [2], en donde la representación de un proceso en espacio de estados sin fallas y perturbaciones está dada por

$$x(k+1) = \mathbf{A}x(k) + \mathbf{B}u(k)\tag{5.2}$$

$$y(k) = \mathbf{C}x(k)\tag{5.3}$$

Al introducir (Ec.5.2) en (Ec.5.3)

$$y(k+1) = \mathbf{C}\mathbf{A}x(k) + \mathbf{C}\mathbf{B}u(k)\tag{5.4}$$

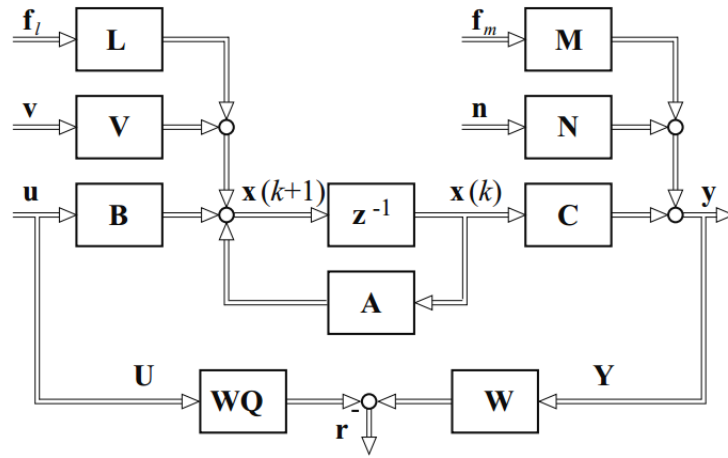


Figura 5.1: Generación de residuos con un modelo en espacio de estados de un sistema MIMO en tiempo discreto.[2]

Obteniendo las siguientes muestras se tiene

$$\begin{aligned} y(k+2) &= \mathbf{C}x(k+2) \\ y(k+2) &= \mathbf{C}\mathbf{A}x(k+1) + \mathbf{C}\mathbf{B}u(k+1) \\ y(k+2) &= \mathbf{C}\mathbf{A}^2x(k) + \mathbf{C}\mathbf{A}\mathbf{B}u(k) + \mathbf{C}\mathbf{B}u(k+1) \end{aligned} \quad (5.5)$$

y para la  $q_{esima}$  muestra se tiene

$$y(k+q) = \mathbf{C}\mathbf{A}^q x(k) + \mathbf{C}\mathbf{A}^{q-1}x(k) + \dots + \mathbf{C}\mathbf{B}u(k+q-1) \quad (5.6)$$

Los pasos anteriores generan ecuaciones redundantes para diferentes instantes de tiempo. La (Ec.5.7) se obtiene al generar ecuaciones, como (5.6), para una ventana de tiempo y al desplazar  $q$  muestras hacia atrás.

$$\mathbf{Y}(k) = \mathbf{T}x(k-q) + \mathbf{Q}\mathbf{U}(k) \quad (5.7)$$

siendo los vectores

$$\mathbf{Y} = \begin{bmatrix} y(k-q) \\ y(k-q+1) \\ \vdots \\ y(k) \end{bmatrix} \quad \mathbf{U} = \begin{bmatrix} u(k-q) \\ u(k-q+1) \\ \vdots \\ u(k) \end{bmatrix} \quad (5.8)$$

y las matrices

$$\mathbf{T} = \begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \mathbf{C}\mathbf{A}^2 \\ \vdots \\ \mathbf{C}\mathbf{A}^q \end{bmatrix} \quad \mathbf{Q} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{C}\mathbf{B} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{C}\mathbf{A}\mathbf{B} & \mathbf{C}\mathbf{B} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}\mathbf{A}^{q-1}\mathbf{B} & \mathbf{C}\mathbf{A}^{q-2}\mathbf{B} & \dots & \mathbf{C}\mathbf{B} & \mathbf{0} \end{bmatrix} \quad (5.9)$$

entonces en la (Ec.5.7) se forma una redundancia temporal. Al no conocer el vector  $x(k - q)$  se multiplica a toda la ecuación por  $\mathbf{w}^T$  y seleccionando

$$\mathbf{w}^T \mathbf{T} = 0 \quad (5.10)$$

se genera una relación entrada-salida y el residuo se puede definir como

$$r(k) = \mathbf{w}^T \mathbf{Y}(k) - \mathbf{w}^T \mathbf{Q} \mathbf{U}(k) \quad (5.11)$$

Las dimensiones de  $\mathbf{w}^T$  son  $1 \times (q+1)r$ , donde  $r$  es el número de salidas. Si el orden de  $\mathbf{A}$  es  $m$ , la matriz  $\mathbf{T}$  tiene un orden  $m \times (q+1)r$ . Según la (Ec.5.10)  $m$  elementos de  $\mathbf{w}^T$  están determinados. Sin embargo, los restantes  $(q+1)r - m$  elementos de  $\mathbf{w}^T$  pueden ser elegidos libremente. Entonces más residuos pueden ser obtenidos al multiplicar la (Ec.5.7) con un vector  $\mathbf{W}$  y haciendo cumplir (5.10).

$$r(k) = \mathbf{W} [\mathbf{Y}(k) - \mathbf{Q} \mathbf{U}(k)] \quad (5.12)$$

La matriz  $\mathbf{W}$  contiene los vectores  $\mathbf{w}^T$  y el resto de elementos son elegidos tal que se puedan generar residuos estructurales. Esto permite diseñar residuos que reacciones de una u otra manera ante la presencia de diferentes fallas en el sistema.

El espacio de paridad consiste en obtener un grupo de ecuaciones que dependan solo de variables medibles y a las cuales se les pueda revisar su paridad. Esto nos permite establecer la consistencia de un modelo con respecto a sus mediciones en una ventana de tiempo. Al aplicar el procedimiento de la Sección.3.3.3, en el sistema representado por la (Ec.5.13), se obtiene que

$$\begin{aligned} x(k+1) &= \mathbf{A}x(k) + \mathbf{B}u(k) + \mathbf{F}_1 f(k) \\ y(k) &= \mathbf{C}x(k) + \mathbf{D}u(k) + \mathbf{F}_2 f(k) \end{aligned} \quad (5.13)$$

$$\mathbf{Y}(k) - \mathbf{Q} \mathbf{U}(k) = \mathbf{T}x(k - q) + \mathbf{L}_{\mathcal{F}} \mathcal{F}(k) \quad (5.14)$$

definiendo:

$$\mathbf{L}_{\mathcal{F}} = \begin{bmatrix} \mathbf{F}_2 & 0 & \dots & 0 \\ \mathbf{C}\mathbf{F}_1 & \mathbf{F}_2 & \dots & 0 \\ \mathbf{C}\mathbf{A}^2\mathbf{F}_1 & \mathbf{C}\mathbf{F}_1 & \dots & 0 \\ \vdots & \vdots & \dots & 0 \\ \mathbf{C}\mathbf{A}^{q-1}\mathbf{F}_1 & \mathbf{C}\mathbf{A}^{q-2}\mathbf{F}_1 & \dots & \mathbf{F}_2 \end{bmatrix} \quad \mathcal{F}(k) = \begin{bmatrix} f(k - q) \\ f(k - q + 1) \\ f(k - q + 2) \\ \vdots \\ f(k) \end{bmatrix} \quad (5.15)$$

Para eliminar la dependencia de las variables de estado se multiplica por  $\mathbf{W}$  tal que  $\mathbf{W}\mathbf{T} = 0$  y se puede obtener una expresión que solo depende de valores medibles. Al escoger una determinada matriz  $\mathbf{W}$  se puede obtener diferentes residuos para diferentes tipos de fallas

$$r(k) = \mathbf{W}(\mathbf{Y}(k) - \mathbf{Q} \mathbf{U}(k)) = \mathbf{W} \mathbf{L}_{\mathcal{F}} \mathcal{F}(k) \quad (5.16)$$



### 5.1.2. Diseño de residuos usando SVD

#### Descomposición en valores singulares

Teniendo una matriz  $G \in \mathbb{R}^{n \times k}$  esta expresada por

$$G = U\Sigma V^T \quad (5.17)$$

donde  $U \in \mathbb{R}^{n \times n}$ ,  $V \in \mathbb{R}^{k \times k}$

$$UU^T = I_{n \times n}, VV^T = I_{k \times k}$$

y para  $n > k$

$$\Sigma = \begin{bmatrix} \text{diag}(\sigma_1, \dots, \sigma_k) \\ 0_{(n-k) \times k} \end{bmatrix}$$

para  $n < k$

$$\Sigma = \begin{bmatrix} \text{diag}(\sigma_1, \dots, \sigma_k) & 0_{(n-k) \times k} \end{bmatrix}$$

siendo  $(\sigma_1, \dots, \sigma_k)$  los valores singulares de  $G$ .

#### Diseño de residuos

Se puede generar residuos, que cumplan con  $\mathbf{W}\mathbf{T} = 0$ , usando el algoritmo planteado en [17]. Los pasos a seguir para el diseño de la matriz  $\mathbf{W}$  son:

1. Determinar el rango de  $\mathbf{Q}$
2. Establecer las dimensiones de  $\mathbf{Q} \in \mathbb{R}^{q(n \times n)}$
3. Obtener las matrices  $\mathbf{Q} = \mathcal{U}\mathcal{S}\mathcal{V}^T$
4. Obtener  $\mathcal{S}\mathcal{V}$  mediante la pre-multiplicación de  $\mathbf{Q}$  por  $\mathcal{U}^T$ , es decir  $\mathcal{U}^T\mathbf{Q}$
5. Los últimos  $qn - \text{Rango}(\mathbf{Q})$  forman una base para la matriz  $\mathbf{W}$ .
6. Se forma  $\mathbf{W}$  mediante  $\mathbf{W} = \begin{bmatrix} 0 & \bar{\mathbf{W}} \end{bmatrix} \mathcal{U}^T$  con  $\bar{\mathbf{W}}$  de dimensiones  $(qn - \text{Rango}(\mathbf{Q})) \times (qn - \text{Rango}(\mathbf{Q}))$

En la Sección.3.3.3 se modelo las fallas a considerar en el modelo de planta propuesto, lo que nos da como resultado la (Ec.5.18).

$$\begin{aligned} x(k+1) &= \mathbf{A}x(k) + \mathbf{B}u(k) + \mathbf{F}_{l1}f_l(k) + \mathbf{F}_{o1}f_o(k) + \mathbf{F}_{a1}f_a(k) \\ y(k) &= \mathbf{C}x(k) + \mathbf{D}u(k) + \mathbf{F}_{l2}f_l(k) + \mathbf{F}_{o2}f_o(k) + \mathbf{F}_{a2}f_a(k) \end{aligned} \quad (5.18)$$

formando las matrices necesarias para generar residuos mediante descomposición de valores singulares, con  $q = 2$ , obtenemos los siguientes resultados para las fugas y obstrucciones:

$$\mathbf{L}_{\mathcal{F}_{li}} = \begin{bmatrix} \mathbf{F}_{l2i} & 0 \\ \mathbf{C}\mathbf{F}_{l1i} & \mathbf{F}_{l2i} \end{bmatrix} \quad \mathbf{L}_{\mathcal{F}_{oj}} = \begin{bmatrix} \mathbf{F}_{o2j} & 0 \\ \mathbf{C}\mathbf{F}_{o1j} & \mathbf{F}_{o2j} \end{bmatrix}$$

en donde

- $\mathbf{F}_{li}$  con  $i = 1, 2, 3$  representa las columnas de la matriz  $\mathbf{F}_l$  de la (Ec.3.12);
- $\mathbf{F}_{oj}$  con  $j = 1, 2, 3, 4, 5$  representa las matrices, que se multiplican a  $f_{oj}$ , en la (Ec.3.14) ;
- los valores  $j = 1, 2$  en  $\mathbf{F}_{oj}$  relacionados con la obstrucción en el tanque 1, los valores  $j = 3, 4$  con la obstrucción en el tanque 2 y  $j = 5$  con la obstrucción en el tanque 3;
- $\mathbf{F}_{l2i} = \mathbf{F}_{o2i} = 0^{3 \times 1}$ ;

la matriz  $\mathbf{Q}$ , con  $\mathbf{Q} = \mathcal{U}\mathcal{S}\mathcal{V}^T$ , junto con  $\mathbf{W}$  están dadas por

$$\mathbf{Q} = \begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{G} \end{bmatrix} \quad \mathbf{W} = [0 \quad \bar{\mathbf{W}}] \mathcal{U}^T, \quad \bar{\mathbf{W}} \in \mathbb{R}^{6 \times 3}$$

EL diseño de matrices  $\mathbf{W}_f$  que eliminen el efecto de una falla se obtienen haciendo cumplir  $\mathbf{W}_f(\mathbf{W}\mathbf{L}_{\mathcal{F}}) = 0$ , teniendo esto en cuenta podemos definir estas matrices como:

Para las fugas

$$\begin{aligned} \mathbf{W}\mathbf{L}_{\mathcal{F}_{li}} &= \mathcal{U}_i \mathcal{S}_i \mathcal{V}_i^T \\ \mathbf{W}_{li} &= [0 \quad \bar{\mathbf{W}}_i] \mathcal{U}_i^T, \quad \bar{\mathbf{W}}_i \in \mathbb{R}^{6 \times 3} \end{aligned}$$

Para las obstrucciones, en las cuales podemos escoger  $j$  talque se tome en cuenta todas las obstrucciones

$$\begin{aligned} \mathbf{W}\mathbf{L}_{\mathcal{F}_{oj}} &= \mathcal{U}_j \mathcal{S}_j \mathcal{V}_j^T \\ \mathbf{W}_{oj} &= [0 \quad \bar{\mathbf{W}}_j] \mathcal{U}_j^T, \quad \bar{\mathbf{W}}_j \in \mathbb{R}^{6 \times 3} \end{aligned}$$

Por ultimo para cumplir con la (Ec.5.16) los valores de  $\mathbf{U}$  y  $\mathbf{Y}$  son

$$\mathbf{Y} = \begin{bmatrix} y(k-2) \\ y(k-1) \\ y(k) \end{bmatrix} \quad \mathbf{U} = \begin{bmatrix} u(k-2) \\ u(k-1) \\ u(k) \end{bmatrix}$$

Se debe tener en cuenta de que el número de residuos calculados por las matrices  $\mathbf{W}_{li}$  y  $\mathbf{W}_{oj}$  son tres por cada matriz, esto por el número de salidas de la planta.

## 5.2. Observadores

La detección de fallas con observadores de estado usa el residuo entre el proceso y el observador como tal. El observador es una versión adaptada del sistema original. La generación de residuos se observa en la Fig.5.2.

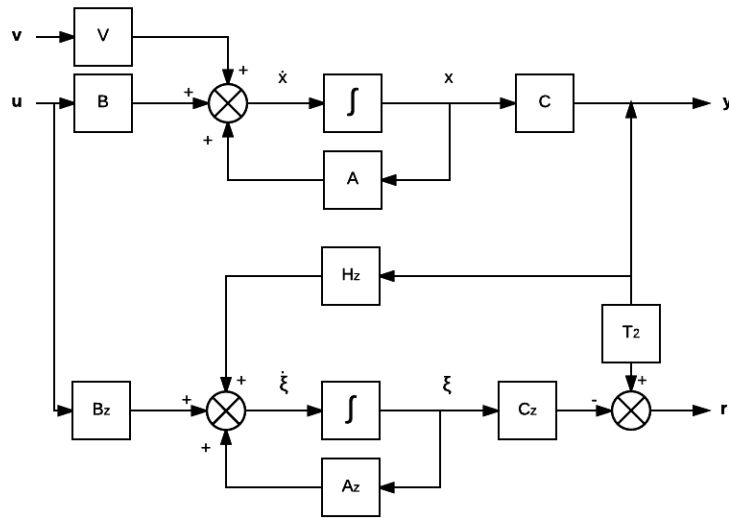


Figura 5.2: Sistema y observador de estado en la salida[2]

Para este trabajo se proponen 3 observadores (uno para cada tanque), con la arquitectura mostrada en la Fig.5.2. Los vectores  $u$  y  $y$  representan las 3 entradas y 3 salidas repectivamente. El vector  $v$  representa las fallas del sistema (desconocidas). El sistema es representado en espacio de estados con las siguientes ecuaciones:

$$\dot{x}(t) = \mathbf{A}x(t) + \mathbf{B}u(t) + \mathbf{V}v(t)$$

$$y(t) = \mathbf{C}x(t)$$

El nuevo estado del observador resulta de una transformación lineal del sistema:

$$\xi(t) = \mathbf{T}_1 \mathbf{x}(t)$$

El estado  $\xi$  del observador se calcula con:

$$\dot{\xi} = \mathbf{A}_z \xi(t) + \mathbf{B}_z u(t) + \mathbf{H}_z y(t)$$

La salida del observador  $\eta$  se calcula con:

$$\eta = \mathbf{C}_z \xi(t)$$

El residuo que se genera con el observador es:

$$\mathbf{r}(t) = \mathbf{T}_2 y(t) - \eta(t)$$

Tomando al error de estado como  $\tilde{\xi}$ :

$$\tilde{\xi} = \xi(t) - \mathbf{T}_1 \mathbf{x}(t)$$

Para desacoplar el error de estado  $\tilde{\xi}$  y el residuo  $r$  se necesitan satisfacer las siguientes ecuaciones:

$$\begin{aligned} \mathbf{T}_1 \mathbf{A} - \mathbf{A}_z \mathbf{T}_1 &= \mathbf{H}_z \mathbf{C} \\ \mathbf{B}_z &= \mathbf{T}_1 \mathbf{B} \\ \mathbf{T}_1 \mathbf{V} &= \mathbf{0} \\ \mathbf{C}_z \mathbf{T}_1 - \mathbf{T}_2 \mathbf{C} &= \mathbf{0} \end{aligned}$$

El diseño del observador es posible si se cumple  $\mathbf{T}_1 \mathbf{V} = \mathbf{0}$ . Se puede utilizar descomposición en valores singulares para que con  $\mathbf{V} = \mathbf{U} \mathbf{S} \mathbf{V}'$  calcular  $\mathbf{T}_1 = [\mathbf{0} \ w] \mathbf{U}'$ .

Las señales del observador no son usadas directamente para la detección de fallas. La función cusum de MATLAB es utilizada para detectar los cambios que se presenten en la media del residuo generado por el observador. Esta función calcula la media y la desviación estándar de un número determinado de muestras; devuelve el punto en donde la suma acumulada es mayor a un umbral de desviaciones estándar de una media objetivo. Se utiliza una ventana de 8 muestras de la tercera derivada de los residuos. Se detecta una falla cuando se supere un umbral de 1.8 de desviación estándar de las muestras enventanadas.

### 5.3. Modelado difuso

Debido a la no linealidad del sistema usado, el modelamiento puede no ser exacto en espacio de estados. El modelamiento difuso es un método eficiente para generar modelos no lineales a partir de una colección de reglas *if-then*. Una técnica de modelamiento difuso, que puede manejar incertidumbre y situación de variación en el tiempo, es el método Takagi-Sugeno (T-S) presentado en [6]. Esta técnica es particularmente buena en el modelamiento de sistemas no lineales mediante un conjunto submodelos difusos. Este modelo es creado con datos medibles de entrada y salida.

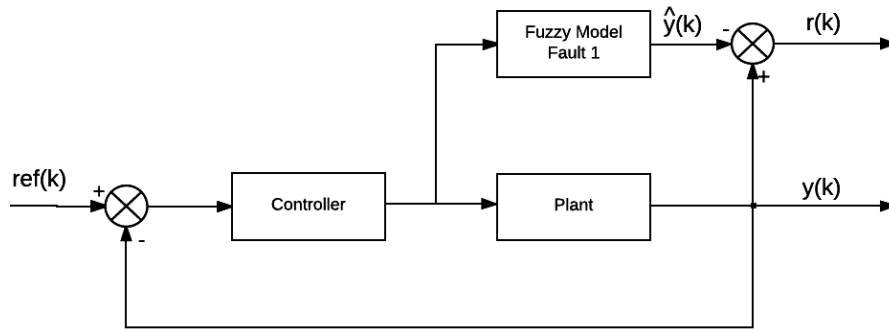


Figura 5.3: Modelamiento e Indentificación Difusa.

El objetivo de usar el modelamiento difuso es comparar su salida con la planta real (Fig. 5.3). Para generar los modelos se realiza un entrenamiento previo de cada uno y se modelan con tres niveles en el valor de las fallas. Los residuos generados se comparan y aquel con menor error indicará el tipo de falla presente en el sistema.

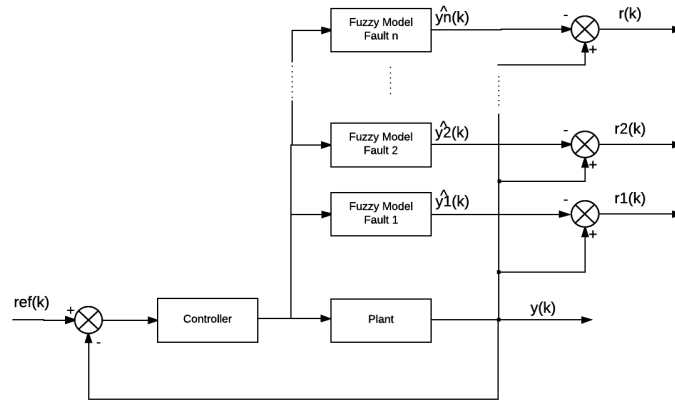


Figura 5.4: Generación de Residuos.

$$r_n(k) = y(k) - \hat{y}_n(k)$$

$$\begin{cases} \min(r(k)) > \lambda & \text{fault detected} \\ r(k) < \lambda & \text{no fault} \end{cases} \quad (5.19)$$

Los modelos son una aproximación de la planta real, por lo cual siempre habrá un pequeño error en el residuo. Para evitar falsas detecciones se establece un umbral  $\lambda$ , de tal manera que solo se detecten fallas reales.

## 5.4. Modelado neuronal

El modelado neuronal se fundamenta en la capacidad de las redes neuronales para realizar ajustes de curva. La idea principal es obtener modelos neuronales en los que la salida de como resultado una estimación del valor que tendría la planta real. La dinámica de los modelos neuronales debe incluir las fallas que se modelan en la sección 3.3.2. El modelo se consigue al realizar un entrenamiento de la red neuronal con valores medibles de la planta. El entrenamiento se debe realizar con datos del modelo real ante la presencia de una falla, esto implica que no se podrá modelar todo el rango de valores posibles de una falla; debido a esto se divide el rango de las fallas en tres: fallas pequeñas, fallas medias y fallas grandes.

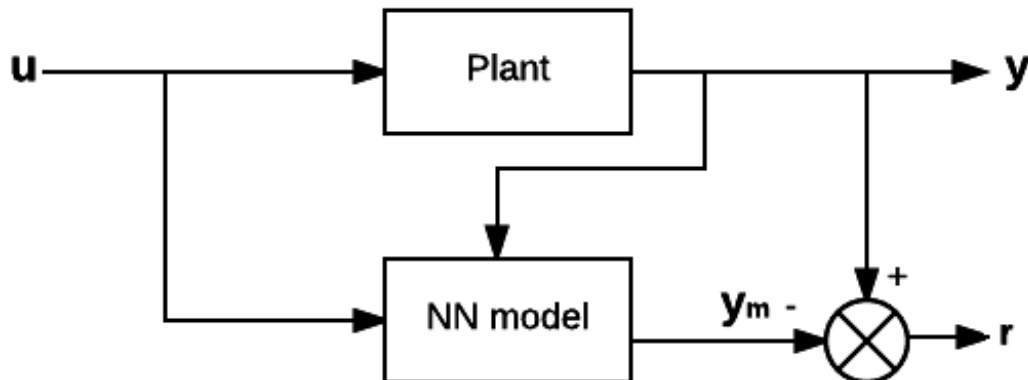


Figura 5.5: Modelamiento neuronal

La arquitectura de la red consiste en la capa de entrada, la capa intermedia y la capa de salida. Las dimensiones de la capa de entrada son determinadas según el número de entradas de la planta, sucede lo mismo que para el caso de las salidas. En cuanto a la capa intermedia, las dimensiones se pueden escoger al azar pero es preferible usar un número que no complique el cálculo computacional. Los valores usados para el entrenamiento son los valores de cada señal de control como entrada y los valores de altura como salidas.

El diagrama de bloques de la Fig. 5.5 muestra la forma de generar residuos. El error, entre la planta y el modelo, será grande hasta el momento que se presente la falla. Cada uno de los residuos reacciona de diferente manera y por ello el que menor error presente indicará el tipo de falla presente.

# Capítulo 6

## Sistema FDD

### 6.1. Detección

Una manera directa de detectar fallas en proceso es comparar el comportamiento del proceso con un modelo del proceso. Este modelo describe el comportamiento nominal, es decir sin fallas en el proceso. Las diferencias de señales entre el proceso y el modelo se reflejan en los residuos, por ende en estos podemos buscar coherencia con las fallas. La detección de una falla es mediante un cambio repentino en el residuo general o en uno de los residuos estructurales. Los métodos, presentados en la sección 5, permiten generar residuos y buscar fallas.

### 6.2. Diagnóstico

El diagnóstico se realiza al comparar residuos diseñados para una determinada falla. Los residuos, generados para fugas y obstrucciones, se comparan para determinar diagnosticar una falla presente (Fig. 6.1). Se selecciona el residuo de menor magnitud en el instante que se detecta una falla en el residuo general.

### 6.3. Estimador de fallas

El estimador, planteado en la Fig.6.2, determina una aproximación de la magnitud de una falla. La información, del tipo de falla y su ubicación, permite generar un modelo post-falla. Un optimizador, mediante redes neuronales, algoritmo de optimización por enjambre de partículas (PSO por sus siglas en inglés), permite reducir el error entre el proceso y el modelo; esto con el fin de aproximar la magnitud de la falla. También se puede obtener la magnitud de la falla al resolver las ecuaciones del modelo post-falla.

#### 6.3.1. Optimizadores

##### Red ADALINE

Una forma de optimizar es mediante la red ADALINE. Esta tiene dos entradas: la primera es el error relativo entre un modelo y la señal medida. La segunda es una re-

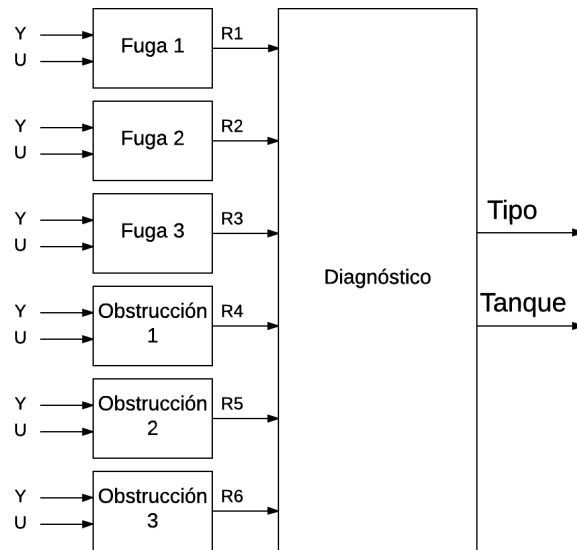


Figura 6.1: Diagrama de bloques del diagnóstico de fallas

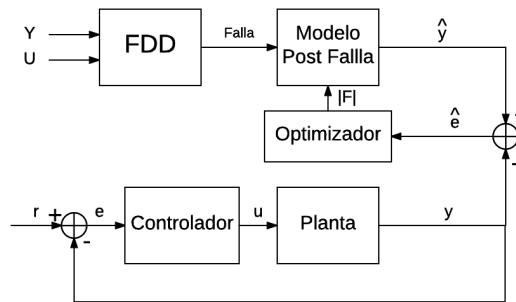


Figura 6.2: Estimador de fallas

troalimentación de la misma red. La red minimiza el error entre el modelo y la salida medida.

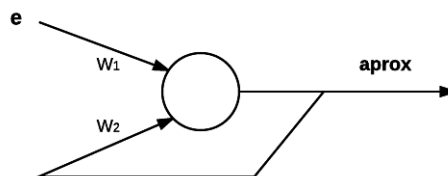


Figura 6.3: ADALINE con una entrada recursiva

## Algoritmo PSO

La función de costo del algoritmo de PSO es el error absoluto entre la señal de referencia del sistema y la respuesta de un modelo del sistema que tiene como variable la falla detectada por el FDD. La posición de cada partícula del enjambre se calcula con la



(Ec.6.1). La velocidad de cada partícula se calcula con la (Ec.6.2).

$$X_i = X_i + \frac{V_i}{1,3} \quad (6.1)$$

$$V_i = \mu V_i + \phi_p r_p (P_i - X_i) + \phi_g r_g (G_i - X_i) \quad (6.2)$$

donde

- $\mu$  representa la inercia.
- $\phi_p$  es el factor de corrección de partícula.
- $\phi_g$  es el factor de corrección de grupo.
- $r_p$  es un valor aleatorio entre 0 y 1, con distribución uniforme.
- $r_g$  es un valor aleatorio entre 0 y 1, con distribución uniforme.
- $P_i$  es la mejor posición conocida de la partícula.
- $G_i$  es la posición de la mejor partícula en el enjambre o grupo.

### 6.3.2. Ecuaciones dinámicas

Las ecuaciones de la sección 3.3.2, mediante métodos numéricos, se resuelven para obtener la magnitud de la falla. Se utilizan los parámetros instantáneos para realizar los cálculos de la falla detectada.

# Capítulo 7

## Control tolerante a fallas

### 7.1. Control Tolerante a Fallas Activo

La detección y diagnóstico de fallas tiene como objetivo robustecer el controlador con información de las fallas del sistema. Los controladores, expuestos en la sección 4, contienen parámetros modificables dependientes del proceso a controlar. Un control tolerante a fallos activo posee la información de la falla y cambia su comportamiento para mantener un rendimiento aceptable. A continuación se presentan dos enfoques para utilizar la información del detector de fallas.

#### 7.1.1. Control MPC con modelo post-falla

Un controlador MPC es robusto ya que calcula los futuros estados del sistema y responde con las señales de control que mejor reducirían su función de costo. El MPC tiene una fuerte dependencia del modelo del sistema a controlar y cuando se presenta una falla se compromete el correcto funcionamiento del control, esto porque el optimizador tiene un modelo del sistema sin fallas. La información de la falla permite generar un modelo post-falla que utiliza el MPC para corregir sus respuestas. El modelo post-falla debe ser lo más preciso posible y solo el tipo de falla y donde se ubica no es suficiente información; entonces un aproximador de la magnitud de la falla mejora el modelo proporcionado al MPC. En la Fig.7.1 se observa un diagrama de bloques explicativo de este controlador.

#### 7.1.2. Controlador de ganancia programada activo

El controlador descrito en la sección 4.4 mejora si la información del detector de fallas es utilizada para cambiar su configuración. Las ganancias de este controlador varían de acuerdo al sistema de ganancia programada difuso. El sistema difuso cambia los valores de ganancia de acuerdo a un factor de corrección, que varía según la magnitud de la falla proporcionada por el aproximador. El factor de corrección es variable y proporcional a la magnitud de la falla. Un gobernador se encarga de cambiar las referencias del sistema si el error persiste por demasiado tiempo. La señal de referencia modificada por el gobernador se define como  $r'_k$ , entonces:

$$r'_k = r'_{k-1} + K_k \cdot [r_k - r'_{k-1}] \quad (7.1)$$

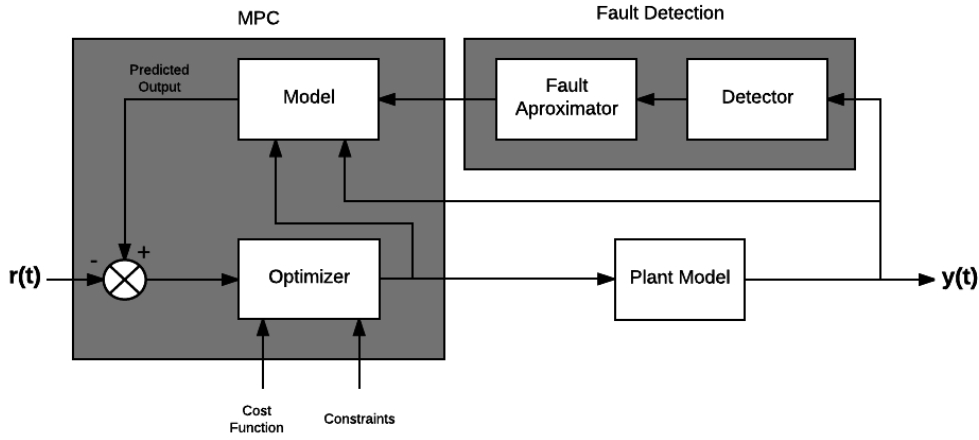


Figura 7.1: MPC activo

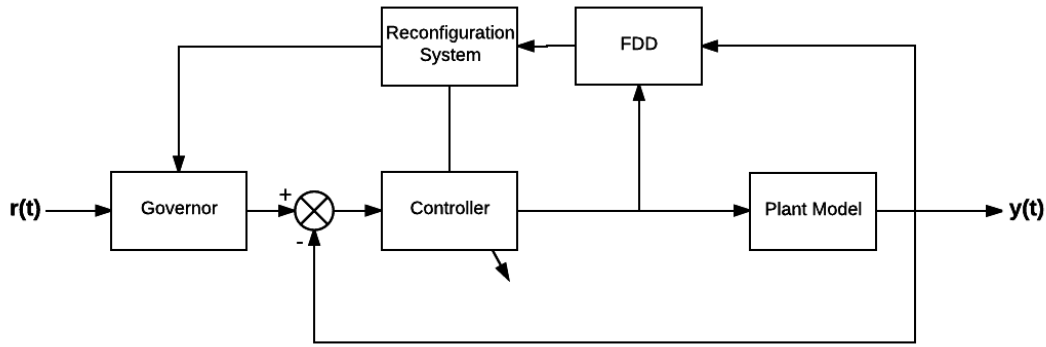


Figura 7.2: Control tolerante a fallas activo

El parámetro  $K_k$  permite regular la transición en los cambios de referencia y así evitar inestabilidades. Si definimos a  $r_k$  como la señal de referencia del sistema y a  $k_D$  como el tiempo en el que ocurre una falla, entonces:

$$r_k = \begin{cases} r_n, & k < k_D \\ r_f, & k \geq k_D \end{cases} \quad y \quad (0 \leq K_k \leq 1) \quad (7.2)$$

El gobernador espera que el error se mantenga durante un tiempo considerable para realizar el cambio de referencia. El tiempo de espera existe para evitar cambios de referencia ante fallas que si pueden ser manejadas por el controlador. Las acciones del gobernador tienen como objetivo mantener la controlabilidad del sistema.

## 7.2. Controlador Tolerante a Fallas Pasivo

Un control tolerante a fallas pasivo (PTFC) no tiene información sobre las fallas que se presentan en un proceso. Este tipo de controlador es diseñado para soportar el efecto de fallas específicas. Los controles adaptativos pueden funcionar como un PTFC, dado que estos se adaptan a los cambios presentes en una planta. El controlador ADALINE PID, presentado en la sección 4.3, es un control adaptativo y se lo compara con un ATFC.

# Capítulo 8

## Simulaciones

### 8.1. Pruebas controlador ADALINE PID

En esta sección se realiza una comparación entre un controlador PID clásico, un controlador adaptativo de modelo de referencia (MRAC) y el controlador ADALINE PID propuesto. El tiempo de muestreo utilizado para las plantas mencionadas en la sección 3 es de  $100ms$ .

En las simulaciones se realizan tres tipos de pruebas: la primera es el rendimiento ante cambios de referencia, la segunda es el rendimiento ante perturbaciones en la salida y la tercera es el cambio de parámetros en el modelo de planta.

El rendimiento de los controladores se mide con el índice de desempeño y el índice de consumo de energía planteados en [18], los cuales están dados por las ecuaciones (8.1) y (8.2).

$$J = \sum_{k=1}^{\infty} \|y(k) - r(k)\|^2 \quad (8.1)$$

$$J = \sum_{k=0}^{\infty} \|e(k)\|^2 + \sum_{k=0}^{\infty} \|u(k)\|^2 \quad (8.2)$$

#### 8.1.1. Planta: Motor CC

En la prueba de cambio de referencia (Fig.8.1) se observa que el controlador propuesto tiene un desempeño similar al PID. Los resultados de rendimiento, índices de desempeño y consumo de energía (Fig.8.2), muestran que el controlador propuesto tiene menor error acumulado y un consumo de energía similar al PID clásico. Los resultados se resumen en la tabla 8.1

Tabla 8.1: Resultados con cambios de referencia en el motor CC.

	C. Propuesto	PID Clásico	MRAC
Sobreimpulso (%)	0	0	49
T. Estabilización (s)	3	3.5	7
I Desempeño	2.01E+06	2.32E+06	1.57E+06
I.Energía	1.19E+07	1.22E+07	1.30E+07

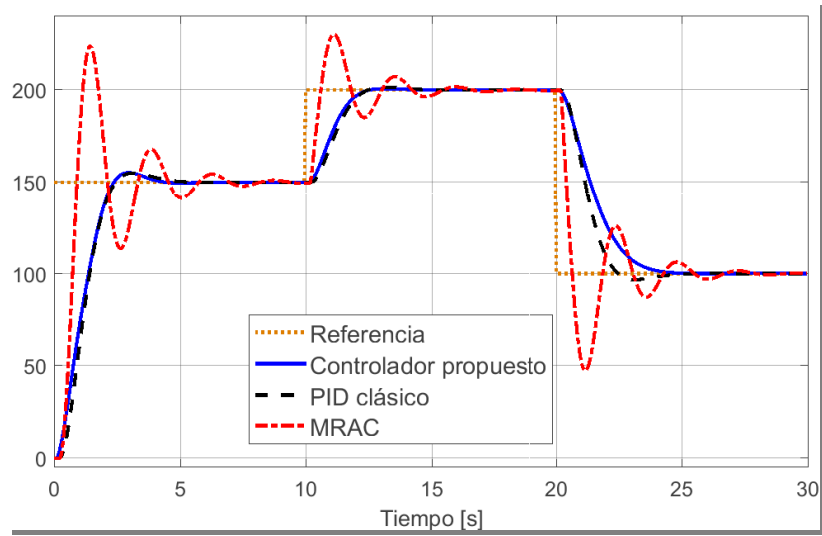


Figura 8.1: Respuesta en lazo cerrado de la planta nominal.

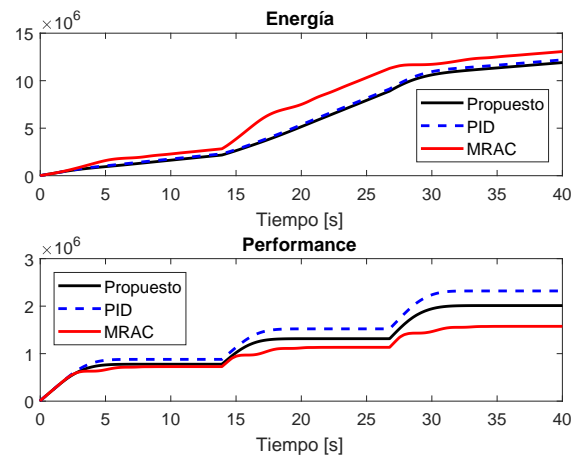


Figura 8.2: Índice de desempeño y consumo de energía de la planta nominal ante cambios de referencia.

Resultados similares se obtienen al realizar las pruebas ante presencia de perturbaciones (Fig.8.3), igualmente el rendimiento se mantiene.

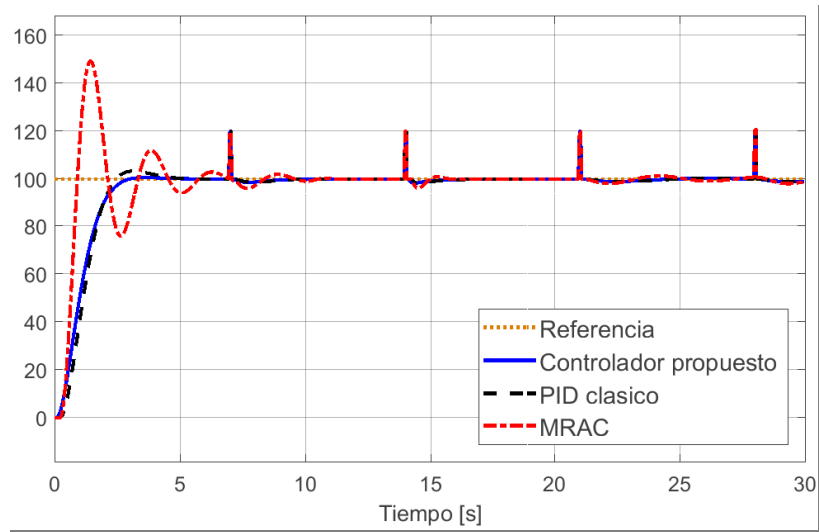


Figura 8.3: Respuesta de la planta nominal ante presencia de perturbaciones.

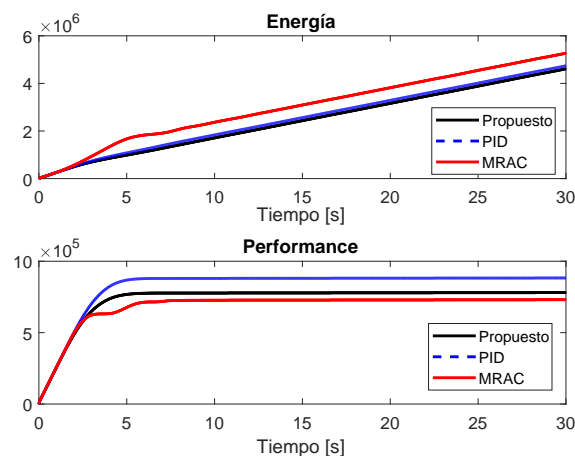


Figura 8.4: Índice de desempeño y consumo de energía de la planta nominal ante presencia de perturbaciones.

Tabla 8.2: Presencia de perturbaciones en el motor CC.

	C. Propuesto	PID Clásico	MRAC
I. Desempeño total	7.80E+05	8.82E+05	7.30E+05
I. Energía total	4.61E+06	4.73E+06	5.27E+06

Los resultados de la tercera prueba realizada, con una variación de  $\pm 50\%$  de la frecuencia natural  $\omega$  y la constante de amortiguamiento  $\zeta$ , se observan en la Fig.8.5. Los cambios de parámetros se dan en el segundo 10 y 20 de la simulación. Primero actúa la planta nominal, luego la planta con  $+50\%$  de variación y finalmente la planta con  $-50\%$ . El error de estado estacionario es el mismo en las tres plantas (Fig.3.1), por ello la referencia cambia en el momento que se cambia el tipo de planta.

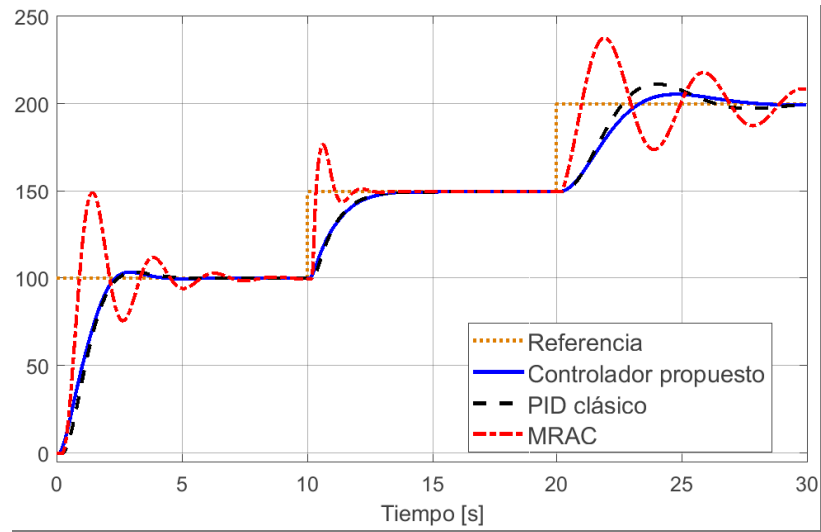


Figura 8.5: Respuesta de la planta con un cambio de planta y referencia en los segundos 10 y 20.

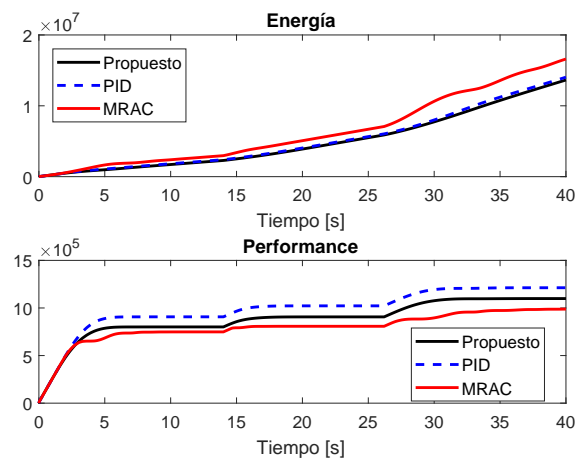


Figura 8.6: Índice de desempeño y consumo de energía ante el cambio de planta y referencia en los segundos 10 y 20.

Tabla 8.3: Variación de parámetros y referencias en el motor CC.

	C. Propuesto		PID Clásico		MRAC	
Variación (%)	+50	-50	+50	-50	+50	-50
Sobreimpulso (%)	0	11	0	23	54	74
T. Estabilización (s)	3.5	9.4	3.5	9.4	2.4	10.8
I. Desempeño	1.09E+06		1.21E+06		0.98E+06	
I. Energía	1.37E+07		1.38E+07		1.64E+07	

Los resultados obtenidos (Fig.8.5) indican que el controlador propuesta se adapta bien ante el cambio de parámetros en la planta, mantiene un buen rendimiento y su consumo de energía es similar al PID clásico.

### 8.1.2. Planta: CSTR

El segundo grupo de pruebas se realiza con el objetivo de probar la adaptabilidad del controlador. Las mismas pruebas, usadas anteriormente, se aplicaron a esta planta. El PID clásico es sintonizado previamente. Los resultados, ante cambios de referencia (Fig.8.7), muestran que el controlador se adapta bien. El índice de desempeño es mejor que los demás controladores y se mantiene el consumo de energía con respecto a los otros controladores (Fig.8.8).

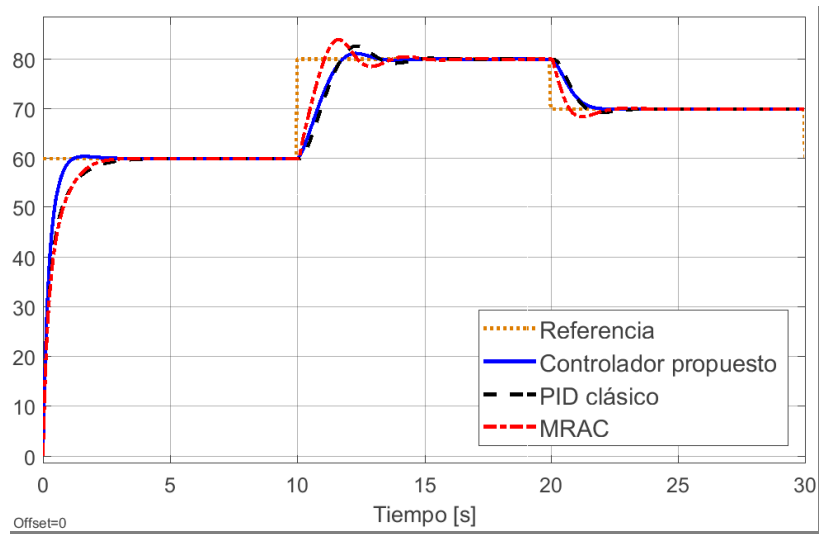


Figura 8.7: Respuesta en lazo cerrado del CSTR nominal.

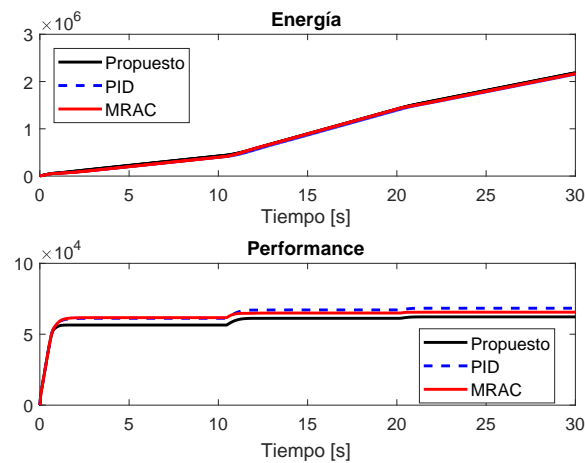


Figura 8.8: Índice de desempeño y consumo de energía del CSTR nominal ante cambios de referencia.

Al ingresar perturbaciones al sistema, se observan resultados similares a los anteriores (Fig.8.9), manteniendo un índice de consumo de energía menor y mejor desempeño.



Tabla 8.4: Resumen de resultados ante cambios de referencia en el CSTR.

	C. Propuesto	PID Clásico	MRAC
Sobreimpulso ( %)	2	3.9	6
T. Estabilización (s)	4.75	3.85	4.45
I. Desempeño	6.09E+04	6.84E+04	6.56E+04
I. Energía	2.17E+06	2.15E+06	2.16E+06

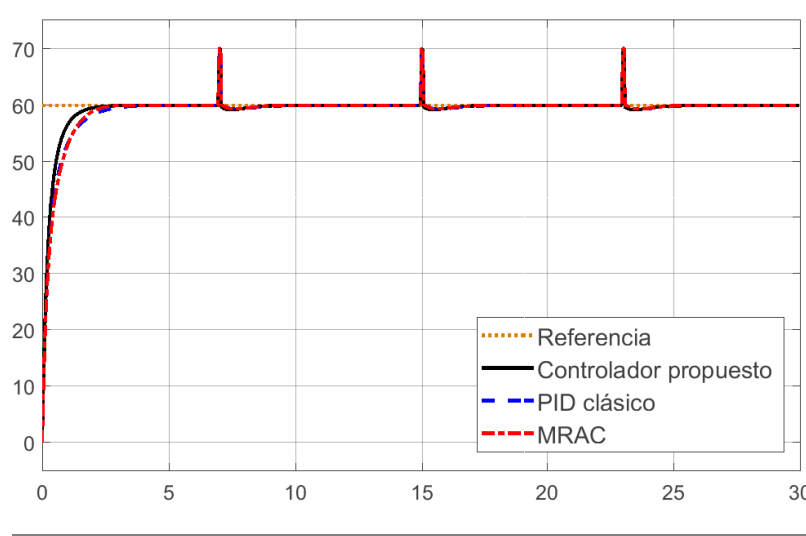


Figura 8.9: Respuesta de la planta nominal ante presencia de perturbaciones.

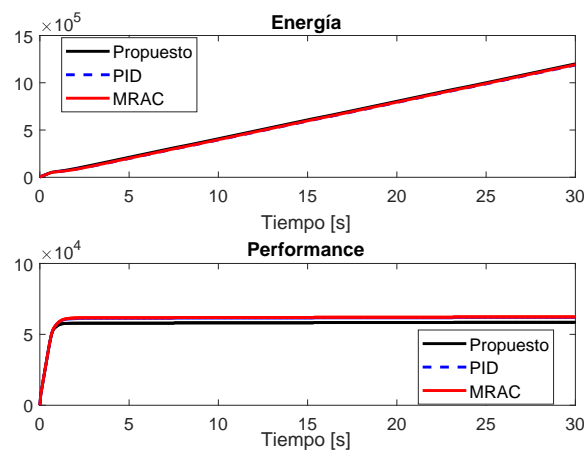


Figura 8.10: Índice de desempeño y consumo de energía de la planta nominal ante presencia de perturbaciones.

Tabla 8.5: Presencia de perturbaciones en el CSTR.

	C. Propuesto	PID Clásico	MRAC
I. Desempeño	5.85E+04	6.18E+04	6.23E+04
I. Energía	1.20E+06	1.10E+06	1.18E+06

Para la prueba de cambio de parámetros se varia el termino de la energía de activación  $\frac{E_a}{R}$  de  $\pm 3\%$  en los segundos 10 y 20 de la simulación (Fig.8.11). Como se esperaba se obtiene un buen desempeño y consumo de energía (Fig.8.12) bajo en otro tipo de planta.

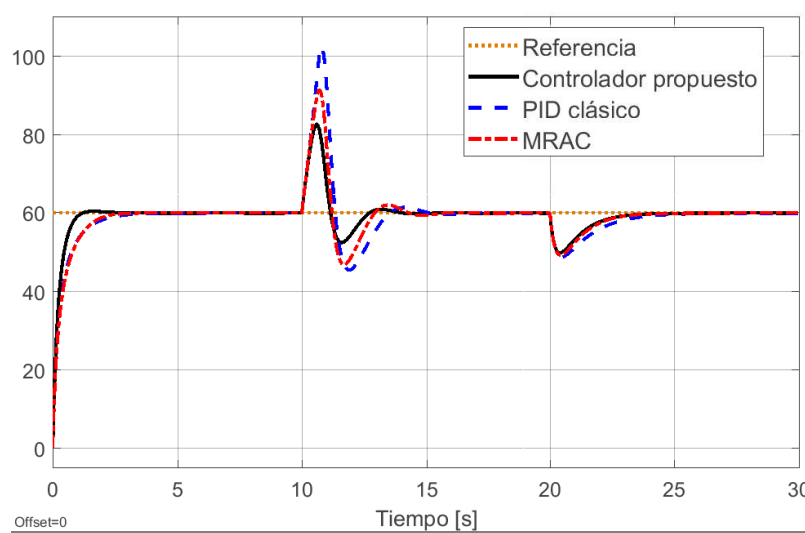


Figura 8.11: Respuesta de la planta con un cambio de planta en los segundos 10 y 20.

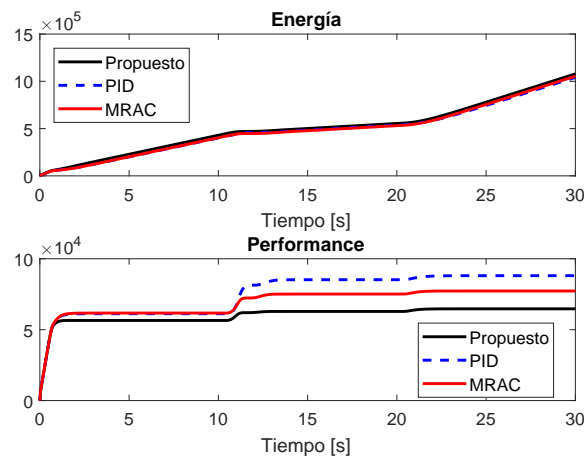


Figura 8.12: Índice de desempeño y consumo de energía ante el cambio de planta en los segundos 10 y 20.

Tabla 8.6: Cambio de parámetros en el CSTR.

	C. Propuesto		PID Clásico		MRAC	
Variación (%)	+3	-3	+3	-3	+3	-3
T. Estabilización(s)	2.7	2.6	3.45	4.4	2.6	3.9
I.Desempeño	6.66E+04		8.80E+04		7.72E+04	
I. Energía	1.05E+06		1.04E+06		1.05E+06	

La ultima prueba ejecutada es la saturación de los actuadores. Esto tiene la finalidad

de comprobar si el efecto de *windup* afecta el comportamiento del controlador. En la Fig.8.13 se comprueba que el controlador no se ve afectado por el efecto *windup*.

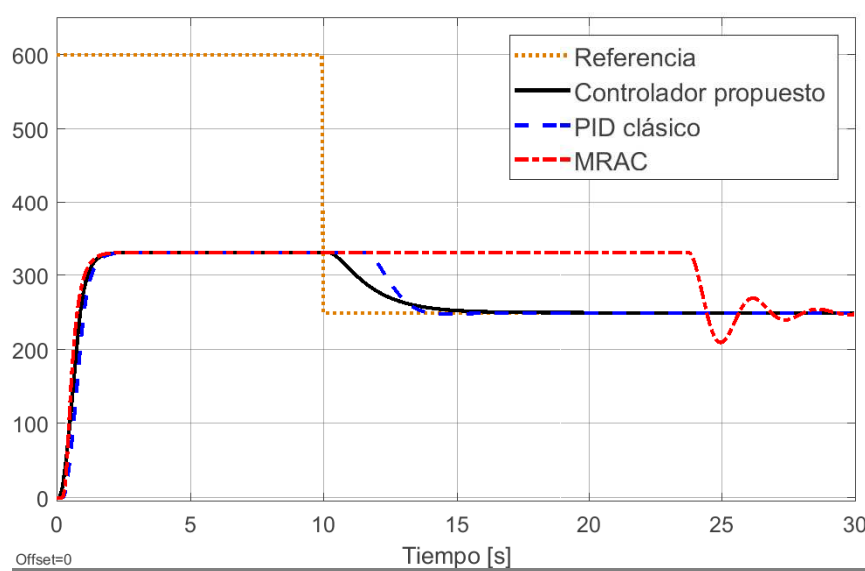


Figura 8.13: Saturación de actuadores en el motor de CC.

En la simulación de la planta propuesta, sistema de 3 tanques interconectados, se usaron las especificaciones de la tabla 8.7. Los parámetros corresponden al modelo planteado en la sección 3.3.1.

Tabla 8.7: Parámetros del sistema de tres tanques interconectados.

Parámetro	símbolo	unidad	Valores	Detalles
Sección del tanque	$S_1, S_2, S_3$	$m^2$		
Sección de las interconexiones	$A$	$m^2$		
Liqueo	$\alpha_1, \alpha_2, \alpha_3$	$m^2$		Agujero en el fondo del tanque
Obstrucción	$\beta_1, \beta_2, \beta_3$	%	0 – 100	Porcentaje de la sección de la interconexión
Flujo de entrada	$f_1, f_3$	$\frac{m^3}{s}$	0 – 0,06	
Válvula en interconexión	$\sigma$	%	0 – 100	porcentaje de apertura
Alturas de líquido	$h_1, h_2, h_3$	$m$	0 – 5	

Las pruebas están divididas en pruebas de los controladores propuestos y pruebas del sistema de detección y diagnóstico de fallas. Las pruebas de los controladores se dividen en pruebas con controladores de base PID y controladores MPC.

## 8.2. Pruebas del sistema de Detección y diagnóstico de Fallas

El sistema de detección y diagnóstico de fallas funciona de manera independiente al controlador. Los diferentes enfoques mencionados en la sección 5 se comparan usando la matriz de confusión. La matriz de confusión permite visualizar el desempeño de un algoritmo de clasificación comparando los valores deseados con los valores obtenidos, en nuestro caso los valores deseados son las fallas ingresadas al sistema y los valores obtenidos son el tipo y ubicación de falla obtenida.

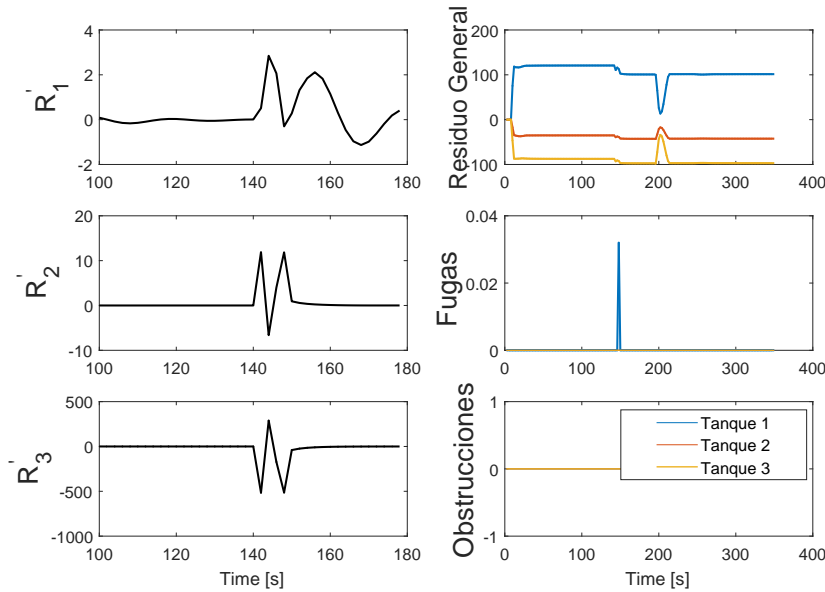


Figura 8.14: Residuos y detección de fallas.

En la Fig.8.14 se puede observar la derivada del residuo general, los residuos estructurales y también la decisión tomada por el bloque FDD. Una falla representa un cambio brusco en el residuo general. Los residuos estructurales reaccionan de la misma manera que el residuo general, pero con diferente magnitud y signo según la falla presente. En el enfoque de observadores se toma el residuo con mayor magnitud para diagnosticar la falla, no así en los demás enfoques toman el residuo con menor magnitud para el diagnóstico.

El rendimiento del enfoque esta determinado por el correcto diagnóstico de las fallas sin importar la magnitud de las misma. Entonces las fallas ingresadas en el clasificador tienen diez posibles valores, que están entre el mínimo y un maximo establecido para cada falla. En total se tiene un set de 10 fallas de liqueo en cada tanque y 10 fallas de obstrucción. Los resultados obtenidos, en la matriz de confusión, se observan en las figuras 8.15, 8.16, 8.18 y 8.17.

Confusion Matrix							
Output Class	1	2	3	4	5	6	
	10 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	10 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	3 5.0%	7 11.7%	0 0.0%	0 0.0%	0 0.0%	70.0% 30.0%
	0 0.0%	0 0.0%	0 0.0%	10 16.7%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 16.7%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 16.7%	100% 0.0%
Target Class							100% 95.0%

Figura 8.15: Matriz de confusión del bloque FDD usando espacios de paridad.

Confusion Matrix							
Output Class	1	2	3	4	5	6	
	10 16.7%	0 0.0%	0 0.0%	1 1.7%	0 0.0%	0 0.0%	90.9% 9.1%
	0 0.0%	10 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	10 16.7%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	9 15.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 16.7%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 16.7%	100% 0.0%
Target Class							
							100% 98.3%

Figura 8.16: Matriz de confusión del bloque FDD usando observadores.

En este caso la obstrucción del tanque 1 con magnitud 0.1 no se detectó al segundo 142 (lo hizo en el 218). La salida fue zero pero en la matriz de confusión se muestra como un falso positivo.

Confusion Matrix							
Output Class	1	2	3	4	5	6	
	9 15.0%	0 0.0%	0 0.0%	1 1.7%	0 0.0%	0 0.0%	90.0% 10.0%
	1 1.7%	8 13.3%	1 1.7%	0 0.0%	0 0.0%	0 0.0%	80.0% 20.0%
	0 0.0%	1 1.7%	7 11.7%	1 1.7%	0 0.0%	1 1.7%	70.0% 30.0%
	10 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0.0% 100%
	10 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0.0% 100%
	0 0.0%	9 15.0%	0 0.0%	0 0.0%	0 0.0%	1 1.7%	10.0% 90.0%
Target Class							
1	2	3	4	5	6		
30.0% 70.0%	44.4% 55.6%	87.5% 12.5%	0.0% 100%	NaN% NaN%	50.0% 50.0%	41.7% 58.3%	

Figura 8.17: Matriz de confusión del bloque FDD usando Modelamiento Neuronal.

Confusion Matrix							
Output Class	1	2	3	4	5	6	
	9 15.0%	0 0.0%	0 0.0%	1 1.7%	0 0.0%	0 0.0%	90.0% 10.0%
	1 1.7%	8 13.3%	1 1.7%	0 0.0%	0 0.0%	0 0.0%	80.0% 20.0%
	0 0.0%	1 1.7%	7 11.7%	1 1.7%	0 0.0%	1 1.7%	70.0% 30.0%
	10 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0.0% 100%
	10 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0.0% 100%
	0 0.0%	9 15.0%	0 0.0%	0 0.0%	0 0.0%	1 1.7%	10.0% 90.0%
Target Class							
1	2	3	4	5	6		
30.0% 70.0%	44.4% 55.6%	87.5% 12.5%	0.0% 100%	NaN% NaN%	50.0% 50.0%	41.7% 58.3%	

Figura 8.18: Matriz de confusión del bloque FDD usando Modelamiento Difuso.

### 8.3. Pruebas de los controladores

Las referencias, de altura del liquido de cada tanque, son los valores  $ref = [3, 2, 1]$  respectivamente. El sistema de generación de residuos funciona normalmente desde el inicio, pero la detección y el diagnóstico se realiza una vez los residuos se hayan estabilizado.

Los controladores tienen las siguientes especificaciones:

### 1. Control PID

Se usan 3 controladores PID con saturación, uno en cada entrada del proceso. Las ganancias proporcionales para cada controlador son  $P = [0,019; 0,0029; 2,33]$  respectivamente, las ganancias integrales son  $I = [0,029; 0,0009; 2,66]$  y todos los controladores tienen una ganancia derivativa de  $T_d = [0,001]$ . Se tiene en cuenta que  $K_p = P$ ,  $T_i = K_p/I$  y  $T_t = \sqrt{(T_i * T_d)}$ .

### 2. Control FGSPID

El controlador de PID de ganancia programada, planteado en la sección 4.4, tiene una configuración similar a un PID. Existen 1 controlador por cada entrada del proceso. La configuración se mantiene si no existen fallas; pero cuando se presenta una falla los valores de sus ganancias se adaptan de acuerdo al sistema difuso. El sistema difuso tiene dos formas de usarse: la primera consiste en establecer una corrección de las ganancias en una proporción fija, es decir los valores de ganancia varían un máximo  $k_v$  % de su valor original, esto a fin de evitar la inestabilidad del sistema. La segunda opción consiste en usar la estimación de la falla, proporcionada por el bloque FDD, para variar  $k_v$ .

### 3. Control ADALINE PID

De manera similar, se usa un controlador por entrada. Este controlador no necesita mayor calibración y solo se debe ajustar el tiempo de muestreo, que para este caso fue de 0,1 segundos.

### 4. Control MPC

Este controlador usa el algoritmo de *Punto Interior* para la optimización. La función de costo proporcionada al controlador es la suma del error cuadrático en las altura de cada tanque. El parámetro de horizonte de predicción se fija en 3. Como restricciones se tienen los límites máximos de saturación de las variables de entrada (Tabla.8.7).

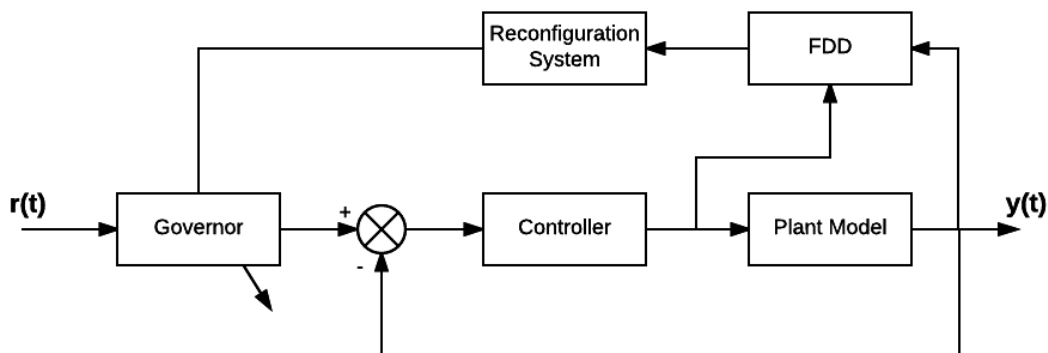


Figura 8.19: Diagrama de bloques para la primera prueba.

Una vez detectada una falla, el sistema difuso se activa para mejorar el tipo de respuesta que tiene el controlador. Luego de una ventana de 40 segundos, si se mantiene el error grande, el gobernador cambia la referencia. El cambio de referencia para liqueos es

reducir las referencias en 20 %, para el caso de obstrucciones es mantener las dos primeras referencias y duplicar la del tercer tanque. La segunda comparación que se realiza es entre el funcionamiento del control MPC sin y con información del modelo post-falla.

En las dos pruebas la magnitud de la falla es lo suficientemente alta como para que el controlador, en funcionamiento normal, no pueda mantener el nivel deseado en los tanques. El diagrama del bloque general usado para la primera prueba se observa en la Fig.8.19, donde el controlador se reemplaza por cada caso anteriormente nombrado. En la segunda prueba se usa el mismo diagrama de bloques para los dos casos del controlador MPC.

### 8.3.1. Comparación controladores PID

La primera comparación es entre el controlador PID, PID de ganancia programada y el controlador ADALINE PID. El controlador FGSPID tiene una variación en el factor de corrección, se prueba con un valor fijo y también con un valor dependiente de la estimación de la magnitud de la falla. Los resultados se observan en las siguientes figuras. Las fallas se presentan en el segundo 142, luego de 40 segundos si el error se mantiene el gobernador realiza un cambio de referencia.

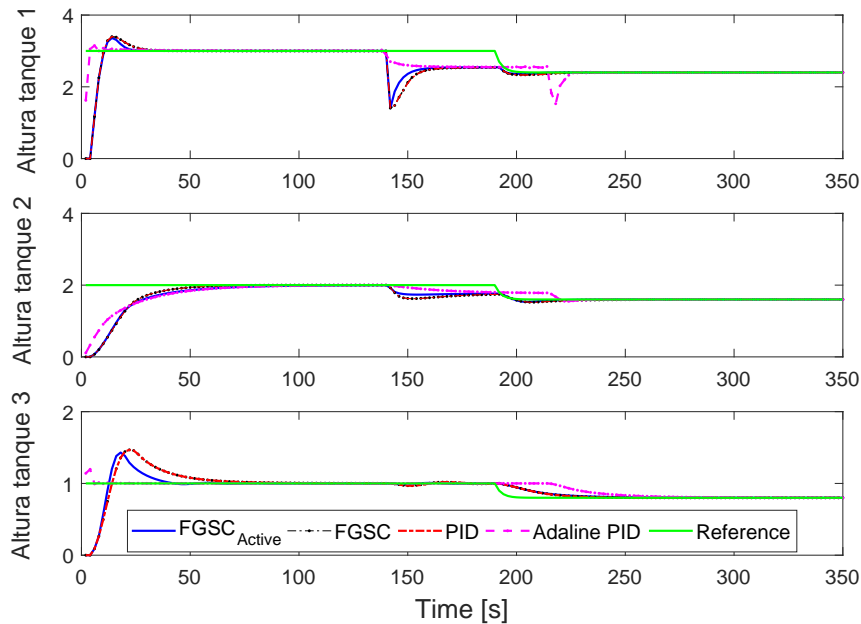


Figura 8.20: Prueba con un liqueo en el primer tanque en el segundo 142.



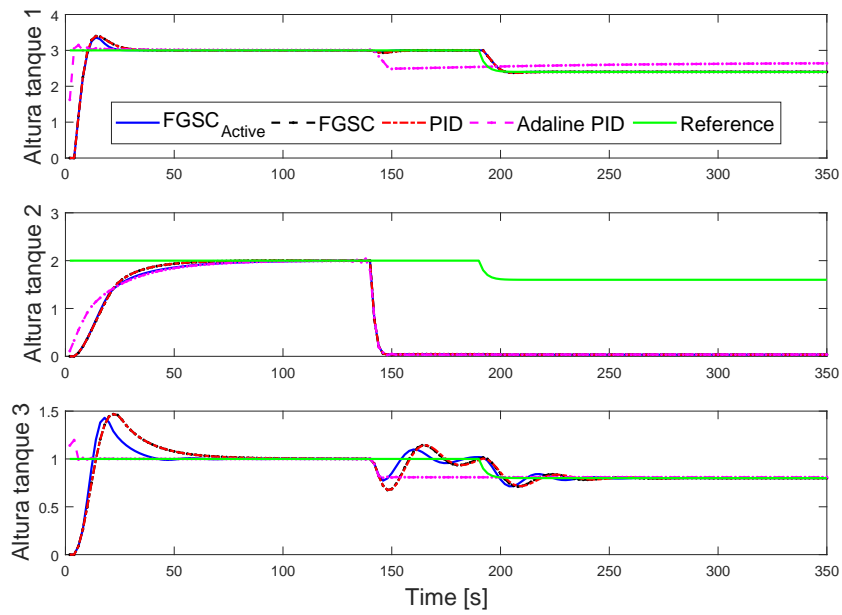


Figura 8.21: Prueba con un liqueo en el segundo tanque en el segundo 142.

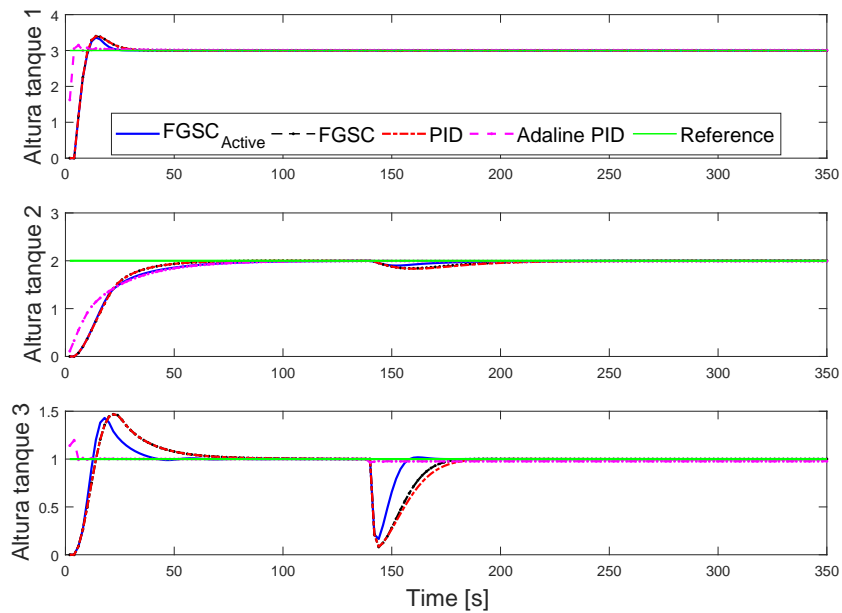


Figura 8.22: Prueba con un liqueo en el tercer tanque en el segundo 142.

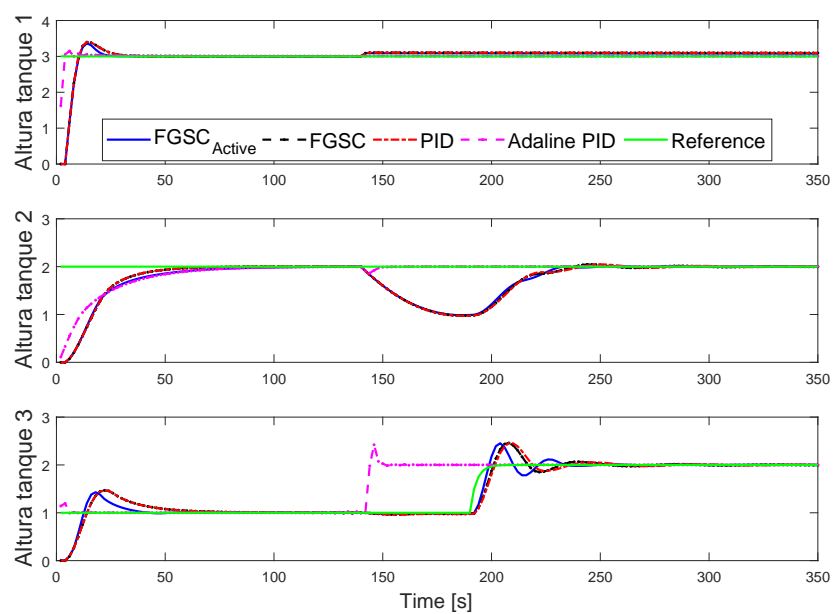


Figura 8.23: Prueba con obstrucción en el primer tanque en el segundo 142.

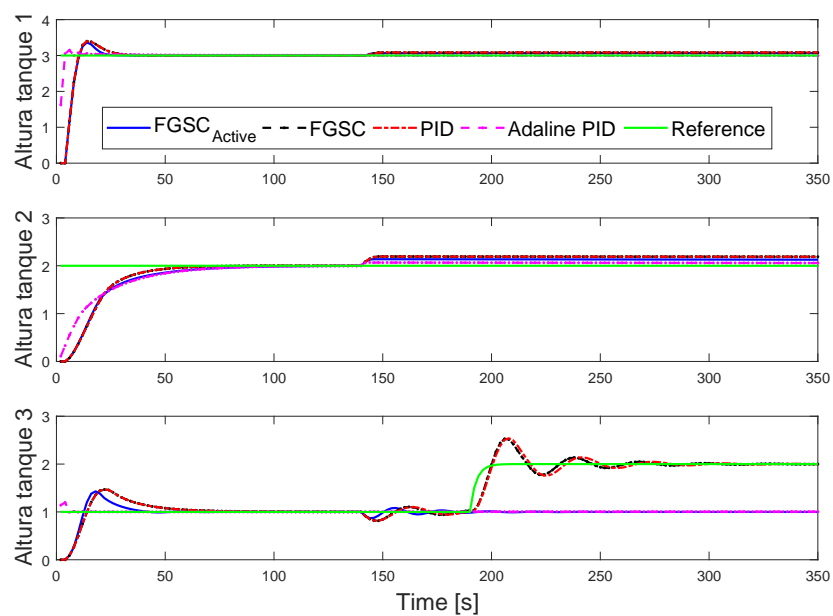


Figura 8.24: Prueba con obstrucción en el segundo tanque en el segundo 142.

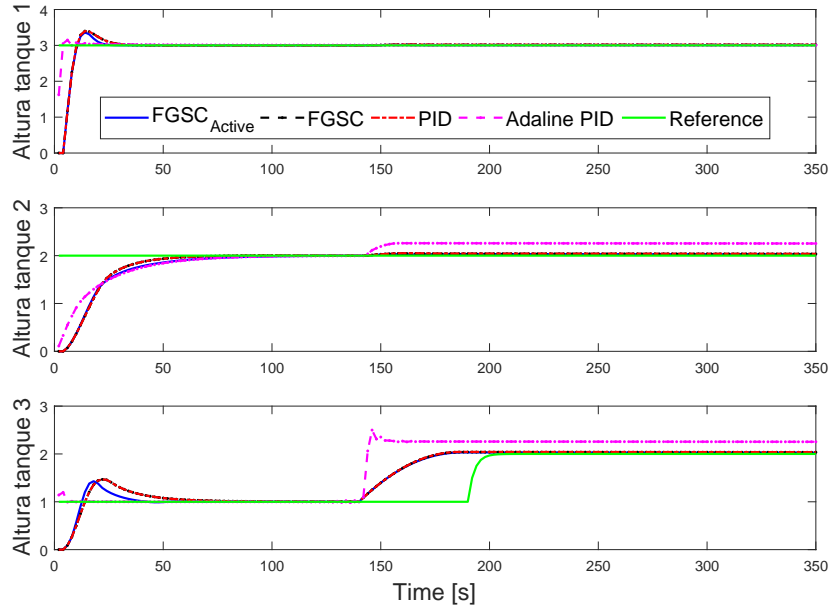


Figura 8.25: Prueba con obstrucción en el tercer tanque en el segundo 142.

La diferencia existente entre los controladores de la prueba anterior es que el control ADALINE no tiene conocimiento de en qué momento se presenta una falla; entonces el gobernador para este se activa cuando el error supera un umbral de 13% y por esto se puede demorar más en activarse.

### 8.3.2. Comparación controlador MPC

El controlador MPC se basa en el modelo de la planta, por ello se compara la respuesta del controlador sin y con modelo de planta post-falla. En las pruebas la falla se presenta en el segundo 141. En estas pruebas no se tiene un gobernador que controle un cambio de referencia después de presentarse una falla. La aproximación de la magnitud de falla tiene una tolerancia de  $\pm 5mm$  en la altura del tanque en el que ocurre la falla. Las fallas de liqueo tienen una magnitud de 0.01 y las obstrucciones se consideran totales.

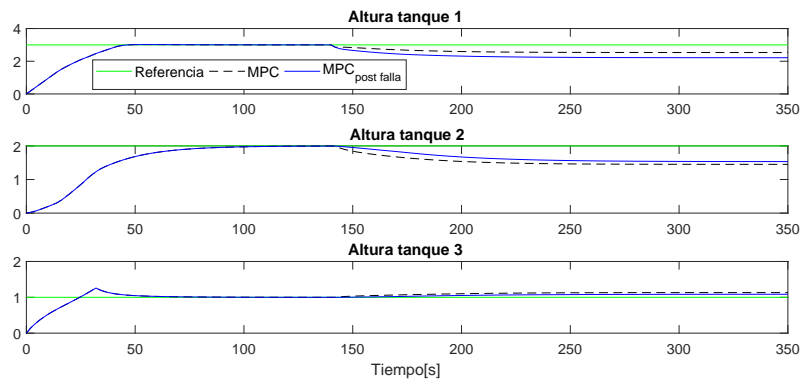


Figura 8.26: Prueba con liqueo en el primer tanque en el segundo 141.

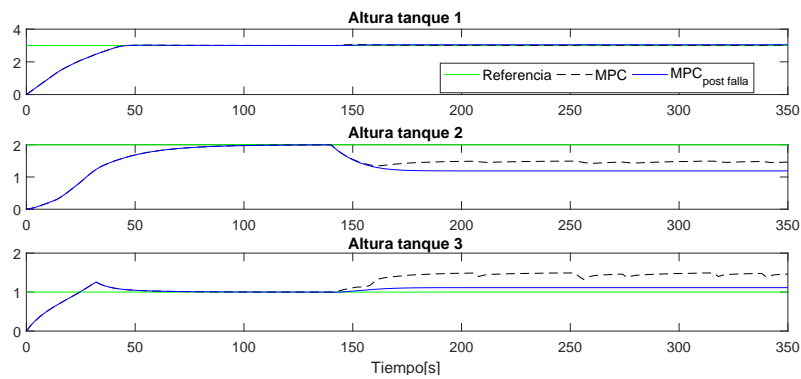


Figura 8.27: Prueba con liqueo en el segundo tanque en el segundo 141.

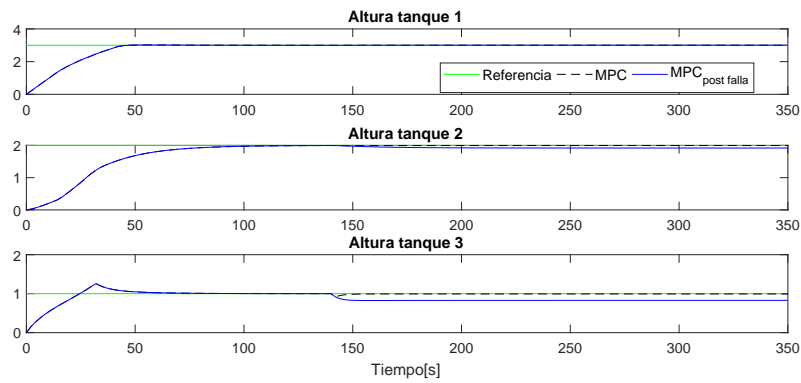


Figura 8.28: Prueba con liqueo en el tercer tanque en el segundo 141.

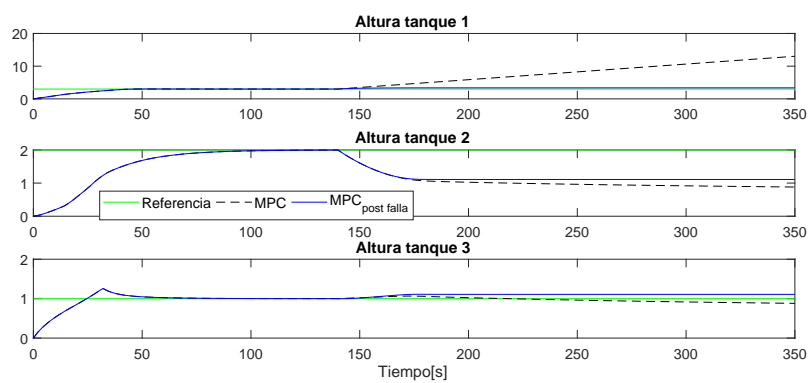


Figura 8.29: Prueba con obstrucción en el primer tanque en el segundo 141.

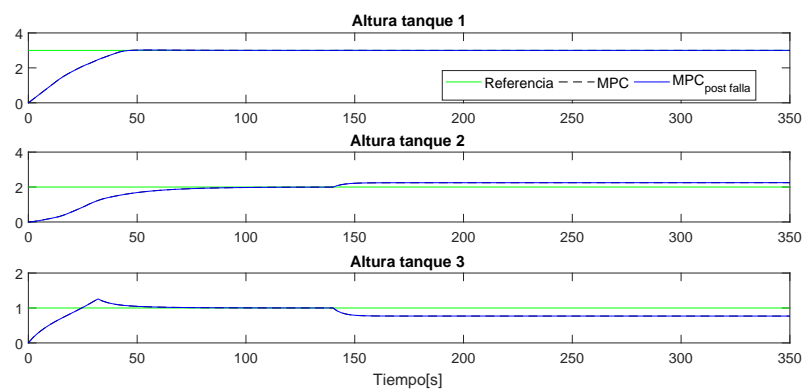


Figura 8.30: Prueba con obstrucción en el segundo tanque en el segundo 141.

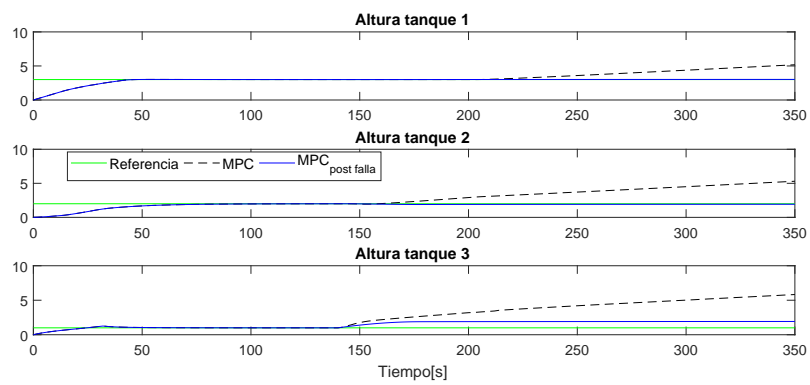


Figura 8.31: Prueba con obstrucción en el tercer tanque en el segundo 141.

# Conclusiones

Un sistema de detección y diagnóstico de fallas proporciona información importante del proceso cuando tiene un comportamiento fuera de lo normal. Esta información es vital cuando se presenta una falla ya que esta puede causar accidentes en el ambiente laboral. La detección de fallas se puede realizar mediante muchos enfoques y con todos se puede obtener buenos resultados. El diagnóstico de fallas implica mayor complejidad. Los resultados son tan buenos como tanto como los modelos empleados precisos. Según la Fig.8.16 el mejor sistema de diagnóstico de fallas puesto a prueba es mediante observadores. Dado que el porcentaje de aciertos con el enfoque de ecuaciones de paridad es alto, este enfoque sería una opción de reemplazo al sistema de observadores.

Un sistema FDD mejora al implementar varios métodos en paralelo y por ende el sistema es más confiable. El modelo de ecuaciones diferenciales de la planta puede ser una desventaja en sistema complejos; pero para estos casos existen otras técnicas de modelamiento más sencillas. El sistema FDD es muy importante en procesos con alto riesgo de accidentes, pero su implementación puede ser costosa; entonces se busca un equilibrio entre costo y confiabilidad.

Los ATFC usan la información dada por el sistema FDD, por ello se espera que su rendimiento sea mejor que el de un PTFC. En las pruebas anteriores se compara un control PID clásico y una variación (FGSPID), buscando que el ATFC mejore su respuesta ante la presencia de una falla. En las figuras 8.20-8.25 se constata un mejor funcionamiento de un ATFC. También al comparar el controlador ADALINE PID con el resto se observa que en ciertos casos, y sin información de las fallas, este tiene mejor tipo de respuesta (Fig. 8.22). Esto nos indica que el controlador ADALINE PID propuesto es una solución alternativa y simple a un ATFC.

Dados los resultados de la sección 8.3.1, en los cuales se comparan controles tolerantes a fallos activos (FGSPID) y pasivos (ADALINE PID), podemos decir que ambos tienen sus ventajas y desventajas en diferentes casos. La elección entre que tipo de control usar depende de la complejidad y confiabilidad que se requiere.

De los resultados mostrados en 8.3.2 se concluye que el MPC no mejora la controlabilidad del sistema como se esperaba, esto se debe a que la aproximación del modelo post-falla tiene una tolerancia. Se observa que el controlador no ofrece una mejora cuando se presentan liqueos. En cuanto a las obstrucciones se nota que las señales de salida dejan de comportarse de manera monótona creciente y se estabilizan pero los resultados de la sección 8.3.1 siguen siendo mejores.

Claramente un control tolerante a fallos presenta mejoras ante controladores más tradicionales, siempre y cuando las fallas que se presentan estén contempladas en el modelo. También la información que se deriva del sistema FDD es otra ventaja, ya que esta serviría para que en un proceso industrial se puedan realizar las reparaciones correspondientes en



la planta. Teniendo en cuenta que las comparaciones arrojaron resultados parejos en ciertos casos, los controladores planteados deben probarse en otros tipos de sistemas MIMO y determinar si los resultados obtenidos en este trabajo se mantienen o varían.

# Bibliografía

- [1] H. Badihi, Y. Zhang, and H. Hong, “Fuzzy gain-scheduled active fault-tolerant control of a wind turbine,” *Journal of the Franklin Institute*, vol. 351, pp. 3677–3706, July 2014.
- [2] R. Isermann, *Fault-Diagnosis Systems an Introduction from Fault Detection to Fault Tolerance*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2006. OCLC: 315813615.
- [3] T. Chaiyatham and I. Ngamroo, “Improvement of power system transient stability by pv farm with fuzzy gain scheduling of pid controller,” *IEEE Systems Journal*, 2014.
- [4] H. Badihi, Y. Zhang, and H. Hong, “Fuzzy gain-scheduled active fault-tolerant control of a wind turbine,” *Journal of the Franklin Institute*, vol. 351, no. 7, pp. 3677–3706, 2014.
- [5] Y. Zhang and J. Jiang, “Fault tolerant control system design with explicit consideration of performance degradation,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 3, pp. 838–848, 2003.
- [6] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, pp. 116–132, Jan. 1985.
- [7] S. Makni, M. Bouattour, M. Chaabane, and A. El Hajjaji, “Robust adaptive observer design for fast fault estimation for nonlinear ts fuzzy systems using descriptor approach,” in *Systems and Control (ICSC), 2017 6th International Conference on*, pp. 249–254, IEEE, 2017.
- [8] Q. Hu, Q. Ren, *et al.*, “The simulation study of adaline pid control and decoupling for the vav system,” in *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM’08. 4th International Conference on*, pp. 1–4, IEEE, 2008.
- [9] A. Dong, Q. Hu, and Q. Ren, “The Simulation Study of Adaline PID Control and Decoupling for the VAV System,” pp. 1–4, IEEE, Oct. 2008.
- [10] J. W. Jung, V. Q. Leu, T. D. Do, E. K. Kim, and H. H. Choi, “Adaptive PID Speed Control Design for Permanent Magnet Synchronous Motor Drives,” *IEEE Transactions on Power Electronics*, vol. 30, pp. 900–908, Feb. 2015.



- [11] B.-y. Xing, L.-y. Yu, and Z.-k. Zhou, “Composite single neural PID controller based on fuzzy self-tuning gain and RBF network identification,” pp. 606–611, IEEE, May 2014.
- [12] K. N, “Studies in modeling and design of controllers for a nonideal continuous stirred tank reactor,” *University*, Aug. 2013.
- [13] B.-Y. Xing, L.-Y. Yu, and Z.-K. Zhou, “Composite single neural pid controller based on fuzzy self-tuning gain and rbf network identification,” in *Control and Decision Conference (2014 CCDC), The 26th Chinese*, pp. 606–611, IEEE, 2014.
- [14] A. Asokan and D. Sivakumar, “Model based fault detection and diagnosis using structured residual approach in a multi-input multi-output system,” *Serbian Journal of Electrical Engineering*, vol. 4, no. 2, pp. 133–145, 2007.
- [15] E. F. Camacho and C. Bordons, *Model predictive control in the process industry*. Springer Science & Business Media, 2012.
- [16] Zhen-Yu Zhao, M. Tomizuka, and S. Isaka, “Fuzzy gain scheduling of PID controllers,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, pp. 1392–1398, Oct. 1993.
- [17] S. X. Ding, *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*. Berlin: Springer, 2008.
- [18] L. I. Minchala-Avila, K. Palacio-Baus, J. P. Ortiz, J. D. Valladolid, and J. Ortega, “Comparison of the performance and energy consumption index of model-based controllers,” in *2016 IEEE Ecuador Technical Chapters Meeting (ETCM)*, pp. 1–6, Oct. 2016.